

# КОНКУРЕНТНО ПРОГРАМИРАНЕ

ВАСИЛ ГЕОРГИЕВ

V.GEORGIEV@FMI.UNI-SOFIA.BG

# СЪДЪРЖАНИЕ

- ПАРАЛЕЛНИ И РАЗПРЕДЕЛЕНИ ПРОЦЕСИ
- ПАРАЛЕЛНИ АЛГОРИТМИ – ПРИНЦИПИ, ПРОЕКТИРАНЕ, МЕТРИКА
- СРЕДИ И ЕЗИЦИ ЗА ПАРАЛЕЛНО ПРОГРАМИРАНЕ
- СИНХРОНИЗАЦИЯ НА ПАРАЛЕЛНИТЕ ПРОЦЕСИ
- ЕТАЛОННИ ПАРАЛЕЛНИ АЛГОРИТМИ
- БАЛАНСИРАНЕ НА ПАРАЛЕЛНАТА ОБРАБОТКА

# ПОСЛЕДОВАТЕЛНИ И ПАРАЛЕЛНИ ПРОГРАМИ

- ПРОГРАМАТА СЕ СЪСТОИ ОТ ПРОЦЕСИ, КОИТО МОГАТ ДА БЪДАТ ИЗПЪЛНЯВАНИ ПОСЛЕДОВАТЕЛНО ИЛИ КОНКУРЕНТНО
- ПРИ ИЗПЪЛНЕНИЕ НА ПРОГРАМА В СРЕДА ЗА ПОСЛЕДОВАТЕЛНОТО ПРОГРАМИРАНЕ
  - ПРОГРАМАТА СЕ СЪСТОИ ОТ ЕДИН ПРОЦЕС
  - РЕЗУЛТАТЪТ ОТ ИЗПЪЛНЕНИЕТО Й С ЕДНАКВИ ДАННИ Е ВИНАГИ ЕДИН И СЪЩ
  - ИЗПЪЛНЕНИЕТО НА ВСЯКА ИНСТРУКЦИЯ Е ПОСЛЕДОВАТЕЛНО И НЕЗАВИСИМО ОТ ИЗПЪЛНЕНИЕТО НА ДРУГИ ИНСТРУКЦИИ
- ПРИ ИЗПЪЛНЕНИЕ НА ПРОГРАМИТЕ В СРЕДИ С МУЛТИПРОГРАМИРАНЕ
  - ПРОГРАМАТА СЕ СЪСТОИ ОТ ЕДИН ПРОЦЕС
  - УПРАВЛЕНИЕТО СЕ ПРЕДАВА ПОСЛЕДОВАТЕЛНО МЕЖДУ РАЗЛИЧНИ ПРОЦЕСИ
  - МЕЖДУ ОТДЕЛНИТЕ ПРОЦЕСИ СЪЩЕСТВУВА ЗАВИСИМОСТ ПО ВРЕМЕ НА ИЗПЪЛНЕНИЕ, НО РЕЗУЛТАТА ОТ ИЗПЪЛНЕНИЕТО ИМ СЕ ЗАПАЗВА
- ПРИ ИЗПЪЛНЕНИЕ НА ПРОГРАМИТЕ В СРЕДИ ЗА ПАРАЛЕЛНО ПРОГРАМИРАНЕ
  - ПРОГРАМАТА СЕ СЪСТОИ ОТ МНОЖЕСТВО ПАРАЛЕЛНИ ПРОЦЕСИ
  - ТЯ ВКЛЮЧВА ОСВЕН УПРАВЛЯВАЩ КОД И ДАННИ, СЪЩО И ИНСТРУКЦИИ ЗА СИНХРОНИЗАЦИЯ И ОБМЕН МЕЖДУ ПРОЦЕСИТЕ, КОИТО СЪСТАВЛЯВАТ НЕЙНИЯ ПЛАНИРАЩ ПРОЦЕС (SCHEDULER)
  - РЕЗУЛТАТЪТ ОТ ИЗПЪЛНЕНИЕТО НА ПАРАЛЕЛНАТА ПРОГРАМА МОЖЕ ДА ЗАВИСИ ОТ РАБОТАТА НА ПЛАНИРАЩИЯ ПРОЦЕС

# ПАРАЛЕЛНИ ПРОЦЕСИ

- ПРОЦЕСИТЕ, ИЗПЪЛНЯВАЩИ ПРОГРАМАТА В СРЕДИТЕ ЗА ПАРАЛЕЛНА ОБРАБОТКА, МОГАТ ДА БЪДАТ АЛТЕРНАТИВНО:
  - РЕПЛИКИ, ИЗПЪЛНЯВАЩИ ЕДНАКВИ ПОДПРОГРАМИ ВЪРХУ РАЗЛИЧНИ ДАННИ – МОДЕЛ SPMD (SINGLE PROGRAM MULTIPLE DATA). N.B.: РАЗЛИКАТА ОТ SIMD Е, ЧЕ В ТОЗИ СЛУЧАЙ СИНХРОНИЗАЦИЯТА СЕ ИЗВЪРШВА НА НИВО ПОДПРОГРАМА (СЕГМЕНТ), А НЕ НА НИВО ИНСТРУКЦИЯ И ЗАТОВА SPMD ОБСЛУЖВАНЕ СЕ ИЗПЪЛНЯВА НА MIMD КОМПЮТРИ
  - РАЗЛИЧНИ ПОДПРОГРАМИ – МОДЕЛ MPMD (MULTIPLE PROGRAM MULTIPLE DATA); ДЕКОМПОЗИЦИЯ НА АЛГОРИТЪМА НА ОТДЕЛНИ ПРОЦЕСИ (ИЛИ НИШКИ) В ТОПОЛОГИЯ, ОТРАЗЯВАЩА ГРАФА НА ЗАВИСИМОСТИ (В 3.5) – НАПРИМЕР ЛИНИЯ (КОНВЕЙЕР) И ДРУГИ СОФТУЕРНИ ШАБЛОНИ (В 5. И 6. ЛЕКЦИЯ)

# ГРАФ НА ПРОЦЕСИТЕ – СКАЛАРНИ ЗАВИСИМОСТИ ([PRECEDENCE | DEPENDENCY] GRAPH)

- ЗАВИСИМОСТТА ПО ДАННИ И УПРАВЛЕНИЕ СЕ ИЗСЛЕДВА [ЧРЕЗ ГРАФИ] НА РАЗЛИЧНИ НИВА – БЛОК, ИЗРАЗ, ПРОМЕНЛИВА
- КОМПИЛАТОРИТЕ ОБИКНОВЕНО ИЗСЛЕДВАТ ГРАФА НА ЗАВИСИМОСТИЕ НА НИВО ИЗРАЗ И ПРОМЕНЛИВА – ПРИМЕР ЗА СЕРИЯТА ИЗРАЗИ (ФИГ. 3.5):

**S1: A = B + C**

**S2: B = A + E**

**S3: A = A + B**

- ИЗРАЗИТЕ СЕ ИЗОБРАЗЯВАТ КАТО ВЪЗЛИ В ГРАФА НА ЗАВИСИМОСТИТЕ, А ДЪГИТЕ СА ЗАВИСИМОСТИТЕ КАТО НАЧАЛОТО НА ДЪГА Е ПРОМЕНЛИВА (АРГУМЕНТ ИЛИ СТОЙНОСТ) НА ИЗРАЗ, А КРАЙ – СЪЩАТА ПРОМЕНЛИВА ОТ СЛЕДВАЩ ИЗРАЗ – ОСВЕН КОГАТО НАЧАЛОТО И КРАЯ НА ДЪГАТА СА АРГУМЕНТИ (ОТ ДЯСНАТА СТРАНА) НА ИЗРАЗИТЕ

# ТИПОВЕ СКАЛАРНИ ЗАВИСИМОСТИ

- **ЗАВИСИМОСТ ПО ДАННИ (DATA FLOW)**: РЕЗУЛТАТА ОТ ИЗРАЗ Е АРГУМЕНТ НА СЛЕДВАЩ ИЗРАЗ (ПРЕНАРЕЖДАНЕТО НА ИЗРАЗИТЕ ИЛИ ПАРАЛЕЛНОТО ИМ ИЗПЪЛНЕНИЕ ПРОМЕНЯ РЕЗУЛТАТА НА СЛЕДВАЩИЯ ИЗРАЗ) — ТАЗИ ЗАВИСИМОСТ Е **НЕПРЕОДОЛИМА**
- **АНТИЗАВИСИМОСТ (ANTI-DEPENDENCY)**: АРГУМЕНТА НА ИЗРАЗ Е РЕЗУЛТАТ ОТ СЛЕДВАЩ ИЗРАЗ (ПРЕНАРЕЖДАНЕТО НА ИЗРАЗИТЕ ИЛИ ПАРАЛЕЛНОТО ИМ ИЗПЪЛНЕНИЕ ПРОМЕНЯ РЕЗУЛТАТА НА АНАЛИЗИРАНИЯ ИЗРАЗ) — ТАЗИ ЗАВИСИМОСТ **МОЖЕ ДА БЪДЕ ПРЕОДОЛЯНА** ЧРЕЗ РЕПЛИКИРАНЕ НА ПРОМЕНЛИВИТЕ
- **ЗАВИСИМОСТ ПО ИЗХОД (DATA OUTPUT)** — РЕЗУЛТАТИТЕ ОТ ДВА ИЗРАЗА СЕ ЗАПИСВАТ В ЕДНА И СЪЩА ПРОМЕНЛИВА (ПРЕНАРЕЖДАНЕ ИЛИ ПАРАЛЕЛНО ИЗПЪЛНЕНИЕ ПРОМЕНЯ СТОЙНОСТТА НА ТАЗИ ПРОМЕНЛИВА) — ТАЗИ ЗАВИСИМОСТ **МОЖЕ ДА БЪДЕ ПРЕОДОЛЯНА** ЧРЕЗ РЕПЛИКИРАНЕ НА ПРОМЕНЛИВИТЕ
- **ЗАВИСИМОСТ ПО ВХОД (DATA INPUT)**: ДВА ИЗРАЗА ИМАТ ОБЩ АРГУМЕНТ — ТАЗИ ЗАВИСИМОСТ НЯМА ЗНАЧЕНИЕ ПРИ СЪВРЕМЕННИТЕ ПРОГРАМНИ СИСТЕМИ (ПОРАДИ СРЕДСТВАТА ЗА КОНКУРЕНТЕН ДОСТЪП)
- **ЗАВИСИМОСТ ПО УПРАВЛЕНИЕ (DATA CONTROL)**: УСЛОВНО ИЗПЪЛНЕНИЕ НА ИЗРАЗ, КЪДЕТО УСЛОВИЕТО Е РЕЗУЛТАТ ОТ ПРЕХОДЕН ИЗРАЗ (ПО СЪЩЕСТВО ТОВА Е РАЗНОВИДНОСТ НА ЗАВИСИМОСТТА ПО ДАННИ)
- ЗА ПО-ВИСОК ПАРАЛЕЛИЗЪМ НА КОДА СЕ ОТСТРАНЯВАТ **АНТИЗАВИСИМОСТИТЕ И ЗАВИСИМОСТИТЕ ПО ИЗХОД**

# ТИПОВЕ ЕДНОМЕРНИ ВЕКТОРНИ ЗАВИСИМОСТИ

- НЕЗАВИСИМИ SPMD

```
for i = 0, 499  
  a[i] = a[i] + C
```

асинхронни  
операции

- ЗАВИСИМИ SPMD

```
for i = 0, 499  
  a[i] = a[mod(i-1|500)] + C
```

аргумент е  
„нова“ стойност

- АНТИЗАВИСИМИ SPMD

```
for i = 0, 499  
  a[i] = a[mod(i+1|500)] + C
```

аргумент е  
„предишна“ стойност

# МОДЕЛИ ОБЩА ПАМЕТ

- В ПАРАЛЕЛНИТЕ СИСТЕМИ ДОСТЪПЪТ ДО ОБЩАТА ПАМЕТ И РЕСУРСИ ЗА В/И Е КОНКУРЕНТЕН И СЕ БАЗИРА НА СХЕМИТЕ ЗА **PRAM** (PARALLEL RANDOM ACCESS MACHINE) — АВТОНОМНИ ПРОЦЕСОРИ С КОНКУРЕНТЕН ДОСТЪП ДО ОБЩА ПАМЕТ (КОЯТО ВКЛЮЧВА И В/И КАНАЛИ)
- В МОДЕЛА PRAM СЕ ПРЕЛАГАТ 4 СХЕМИ ЗА ОТСТРАНЯВАНЕ НА КОНФЛИКТЕН КОНКУРЕНТЕН ДОСТЪП ДО ОБЩОТО АДРЕСНО ПРОСТРАНСТВО:
  - **EREW** (EXCLUSIVE READ, EXCLUSIVE WRITE) — РЕЗЕРВИРАНЕ НА КОНКУРЕНТНИЯ ДОСТЪП ДА ДАДЕН АДРЕС ЗА ДВАТА ТИПА ОПЕРАЦИИ
  - **CREW** (CONCURRENT READ, EXCLUSIVE WRITE) — НЯКОЛКО ПРОЦЕСОРА МОГАТ ДА ЧЕТАТА ЕДНОВРЕМНО ДАДЕН АДРЕС, НО ОПЕРАЦИИТЕ ЗА ЗАПИС СА МОНОПОЛНИ
  - **ERCW** (EXCLUSIVE READ, CONCURRENT WRITE) — ДОПУСКАТ СЕ НЯКОЛКО ЕДНОВРЕМЕННИ ОПЕРАЦИИ НА ЗАПИС НО МОНОПОЛНО ЧЕТЕНЕ
  - **CRCW** (CONCURRENT READ, CONCURRENT WRITE) — КОНКУРЕНТНИТЕ ОПЕРАЦИИ СА БЕЗ ОГРАНИЧЕНИЕ
- **\*\*EW** СХЕМИТЕ СЪОТВЕТСТВАТ НА ИЗИСКВАНИЯТА ЗА КОНСИСТЕНТНОСТ (СЪГЛАСУВАНОСТ И ДЕТЕРМИНИСТИЧНОСТ) НА ДАННИТЕ И СЕ ПРИЛАГАТ КАТО УНИВЕРСАЛНИ ПРИ ПОВЕЧЕТО ПАРАЛЕЛНИ АЛГОРИТМИ;
- КОНКУРЕНТНИТЕ ОПЕРАЦИИ ЗА ЗАПИС ПРИ **\*\*CW** СХЕМИТЕ ИМАТ ОГРАНИЧЕНО ПРИЛОЖЕНИЕ ПРИ НЯКОИ КЛАСОВЕ ПАРАЛЕЛНИ АЛГОРИТМИ ЗА ОБРАБОТКА НА ГРАФИ И ЧИСЛОВА ОБРАБОТКА, ПРИ КОИТО ПОСТИГАТ ПО-ВИСОКО БЪРЗОДЕЙСТВИЕ ОТ СХЕМИТЕ С РЕЗЕРВИРАН ЗАПИС



# МОДЕЛ С ОБМЕН НА СЪОБЩЕНИЯ

- ПРИ ОБМЕН НА СЪОБЩЕНИЯ ВСЯКА ДВОИКА ПРОЦЕСИ Е СВЪРЗАНА С КОМУНИКАЦИОНЕН КАНАЛ, ПРЕДСТАВЕН С ТОЧНО ЕДНА ПРОМЕНЛИВА — ПОСЛЕДОВАТЕЛНИТЕ СЪОБЩЕНИЯ СА СТОЙНОСТИТЕ НА ТАЗИ ПРОМЕНЛИВА;
- ДЕФИНИРАНО Е СЪСТОЯНИЕ НА КАНАЛА — НАПР. ЧЕТЕНЕ НА ПРОМЕНЛИВАТА-КАНАЛ СЕ ДОПУСКА САМО КОГАТО СЪСТОЯНИЕТО МУ НЕ Е ПРАЗЕН (РЕСП. ПРИ ЗАПИС — ДА НЕ Е ПЪЛЕН);
- АСИНХРОННИЯТ И СИНХРОННИЯТ КАНАЛ СА С ЕДНАКЪВ РЕЖИМ НА ДОСТЪП НО АСИНХРОННИЯТ ИМА КАПАЦИТЕТ = РАЗМЕРА НА БУФЕРА ( $>1$ )

# ПАРАЛЕЛНИ АЛГОРИТМИ

- ПАРАЛЕЛНИТЕ АЛГОРИТМИ СА МЕЖДИННОТО ЗВЕНО ВЪВ ВЕРИГАТА НА ПАРАЛЕЛНАТА ОБРАБОТКА (МЕЖДУ ИЗЧИСЛИТЕЛНИЯ ПРОБЛЕМ И ПАРАЛЕЛНАТА СИСТЕМА) –
  - АРХИТЕКТУРА
  - СИСТЕМА/СРЕДА
  - ПРОГРАМА
  - АЛГОРИТЪМ
  - ИЗЧИСЛИТЕЛЕН ПРОБЛЕМ
- ПА Е АБСТРАКТНО (ФОРМАЛНО ИЛИ НЕФОРМАЛНО) ПРЕДСТАВЯНЕ НА ИЗЧИСЛИТЕЛЕН ПРОБЛЕМ КАТО НАБОР ОТ ПРОЦЕСИ ЗА ЕДНОВРЕМЕННО ИЗПЪЛНЕНИЕ (В СЛУЧАЯ ПРОЦЕС Е ЧАСТ ОТ ПРОБЛЕМА, КОЯТО СЕ ИЗПЪЛНЯВА ОТ ЕДИН ПРОЦЕСОР)
- ОСНОВНИТЕ ХАРАКТЕРИСТИКИ НА ПА (КОИТО ОТСЪСТВАТ ПРИ ПОСЛ. АЛГОРИТМИ) СА
  - БРОЙ ПРОЦЕСИ И ЛОГИЧЕСКАТА ИМ ОРГАНИЗАЦИЯ (НАПР. MASTER-SLAVE)
  - РАЗПРЕДЕЛЕНИЕ НА ДАННИТЕ (ДЕКОМПОЗИЦИЯ + ВЪЗМОЖНОСТИ ЗА РАЗПРЕДЕЛЕНА АЛОКАЦИЯ)
  - ТОЧКИ НА СИНХРОНИЗАЦИЯ (ОПТИМИЗИРАНЕ)
  - МОДЕЛ НА МЕЖДУПРОЦЕСНИЯ ОБМЕН (ОСНОВНО ОБЩА ПАМЕТ – ОБМЕН НА СЪОБЩЕНИЯ)
- РАЗЛИЧНИТЕ КОНКРЕТНИ РЕШЕНИЯ НА ГОРНИТЕ ХАРАКТЕРИСТИКИ ПОРАЖДАТ ЦЯЛ КЛАС ОТ ПА, БАЗИРАНИ НА ЕДИН ПОСЛЕДОВАТЕЛЕН АЛГОРИТЪМ

# ФАЗИ НА ПРОЕКТИРАНЕТО НА ПА

- ПРОЕКТИРАНЕТО НА ПА МИНАВА ПРЕЗ СЛЕДНИТЕ ФАЗИ (3.14 ):
  - **РАЗДЕЛЯНЕ** (PARTITIONING) – ДЕКОМПОЗИЦИЯ НА ПРОБЛЕМА:
    - ПО ДАННИ (главно SPMD) или
    - ПО ФУНКЦИИ (главно MPMD) –
    - РАЗДЕЛЯНЕТО СЕ ИЗВЪРШВА С ОГЛЕД НА СПЕЦИФИКАТА НА ПРОБЛЕМА; ЦЕЛТА Е ДА СЕ ДЕФИНИРАТ МНОЖЕСТВО ПОДЗАДАНИЯ; ГРАНУЛАРНОСТТА ПРИ ТАЗИ ФАЗА НЕ ОТЧИТА ОСОБЕНОСТИТЕ НА АРХИТЕКТУРАТА, КОЯТО ЩЕ СЕ ИЗПОЛЗВА ЗА ОБРАБОТКА – РЕЗУЛТАТЪТ ОТ ФАЗАТА Е ДЕФИНИЦИЯ НА ОТДЕЛНИТЕ ЗАДАНИЯ
  - **КОМУНИКАЦИИ** (И ЗАВИСИМОСТ) (COMMUNICATION) – ФОРМУЛИРА ИНФОРМАЦИОННИТЕ ИЛИ КОНТРОЛНИТЕ ЗАВИСИМОСТИ МЕЖДУ ОТДЕЛНИТЕ ПОДЗАДАНИЯ; КОМУНИКАЦИИТЕ СЕ ПРЕДСТАВЯТ КАТО КАНАЛИ (СЪС СЪОТВЕТНИТЕ СВОЙСТВА – НАПР. КАПАЦИТЕТ, ПОСОКА) И СЪОБЩЕНИЯ (Т.Е. ДАННИ И КОМАНДИ), КОИТО СЕ ПРЕДАВАТ ПО ТЕЗИ КАНАЛИ (НАПР. ФОРМАТ, РАЗМЕР, ТИП); АРХИТЕКТУРАТА ЗА ОБРАБОТКА СЕ ИГНОРИРА И НА ТАЗИ ФАЗА, НО СПЕЦИФИЦИРАНЕТО НА КАНАЛИТЕ ПОМАГА ДА СЕ ОЦЕНИ АЛГОРИТЪМА ПО КОМУНИКАЦИОННА СЛОЖНОСТ

# ... ФАЗИ НА ПРОЕКТИРАНЕТО НА ПА

- **ФОРМИРАНЕ** (AGGLOMERATION) — СЛЕД ОЦЕНКА НА ИЗЧИСЛИТЕЛНАТА И КОМУНИКАЦИОННАТА СЛОЖНОСТ НА ФОРМУЛИРАНИТЕ ПОДЗАДАНИЯ И ПРИЛЕЖАЩИТЕ ИМ КОМУНИКАЦИИ, ТЕ СЕ ГРУПИРАТ В ЗАДАНИЯ, ПРИ КОЕТО СЕ ОТЧИТАТ ХАРАКТЕРИСТИКИТЕ НА АРХИТЕКТУРАТА НА ОБРАБОТКА — ОСНОВНО БРОЙ ПРОЦЕСОРИ/ВЪЗЛИ И КОМУНИКАЦИОНЕН МОДЕЛ — И В РЕЗУЛТАТ СЕ ПОСТИГА ОПТИМИЗИРАНЕ ПО СЛЕДНИТЕ ХАРАКТЕРИСТИКИ
  - ГРАНУЛАРНОСТ И БАЛАНСИРАНОСТ (С ОЦЕНКА НА ИЗЧИСЛИТЕЛНАТА СЛОЖНОСТ НА ОТДЕЛНИТЕ ЗАДАНИЯ)
  - ЕВЕНТУАЛНО РЕПЛИКИРАНЕ НА ДАННИ И ПОДЗАДАНИЯ
  - ОПТИМИЗИРАНЕ НА КОМУНИКАЦИИТЕ (С ОЦЕНКА НА КОМУНИКАЦИОННАТА СЛОЖНОСТ НА ОТДЕЛНИТЕ ЗАДАНИЯ)
  - ЕВЕНТУАЛНО ЗАПАЗВАНЕ НА ЛИНЕЙНОСТ (СКАЛИРУЕМАСТ)
  - ТЕХНОЛОГИЧНО ОПТИМИЗИРАНЕ (НАПР. НАМАЛЯВАНЕ НА РАЗХОДИТЕ ЗА КОДИРАНЕ НА ЗАДАНИЯТА)

# ... ФАЗИ НА ПРОЕКТИРАНЕТО НА ПА

- **РАЗПРЕДЕЛЯНЕ** (MAPPING) – НЕЗАДЪЛЖИТЕЛНА ФАЗА (ОТСЪСТВА ПРИ ПРОЕКТИРАНЕ НА ПА ЗА СИСТЕМИ С ДИНАМИЧНО ПЛАНИРАНЕ – ОБИКН. МП С РОС), КОЯТО СЕ СЪСТОИ В РАЗПРЕДЕЛЯНЕ НА ФОРМИРАНИТЕ ЗАДАНИЯ (ИЛИ ЕВЕНТУАЛНО ГРУПИ ОТ ЗАДАНИЯ) ПО ОБРАБОТВАЩИТЕ ВЪЗЛИ НА СИСТЕМАТА СЪС КОДИРАНЕ НА СЪОТВЕТНОТО РЕШЕНИЕ. **N.B.:** ОБИКНОВЕНО СЕ ИЗПОЛЗВА СПЕЦИАЛЕН ЕЗИК ЗА СПЕЦИФИКАЦИЯ НА ЗАРЕЖДАНЕТО И ЕВЕНТУАЛНО ЗА НАСТРОКА НА КОМУНИКАЦИОННИТЕ КАНАЛИ НАПР. В СИСТЕМИ С КОМУТИРУЕМИ КАНАЛИ, ТАКА ЧЕ ОТ АЛГОРИТЪМА СЕ ИЗИСКВА ДА СПЕЦИФИЦИРА И КОМУНИКАЦИОННИЯ ГРАФ НА СИСТЕМАТА ЗА ОБРАБОТКА 5.1

# МЕТРИКА И АНАЛИЗ НА ПРОИЗВОДИТЕЛНОСТА

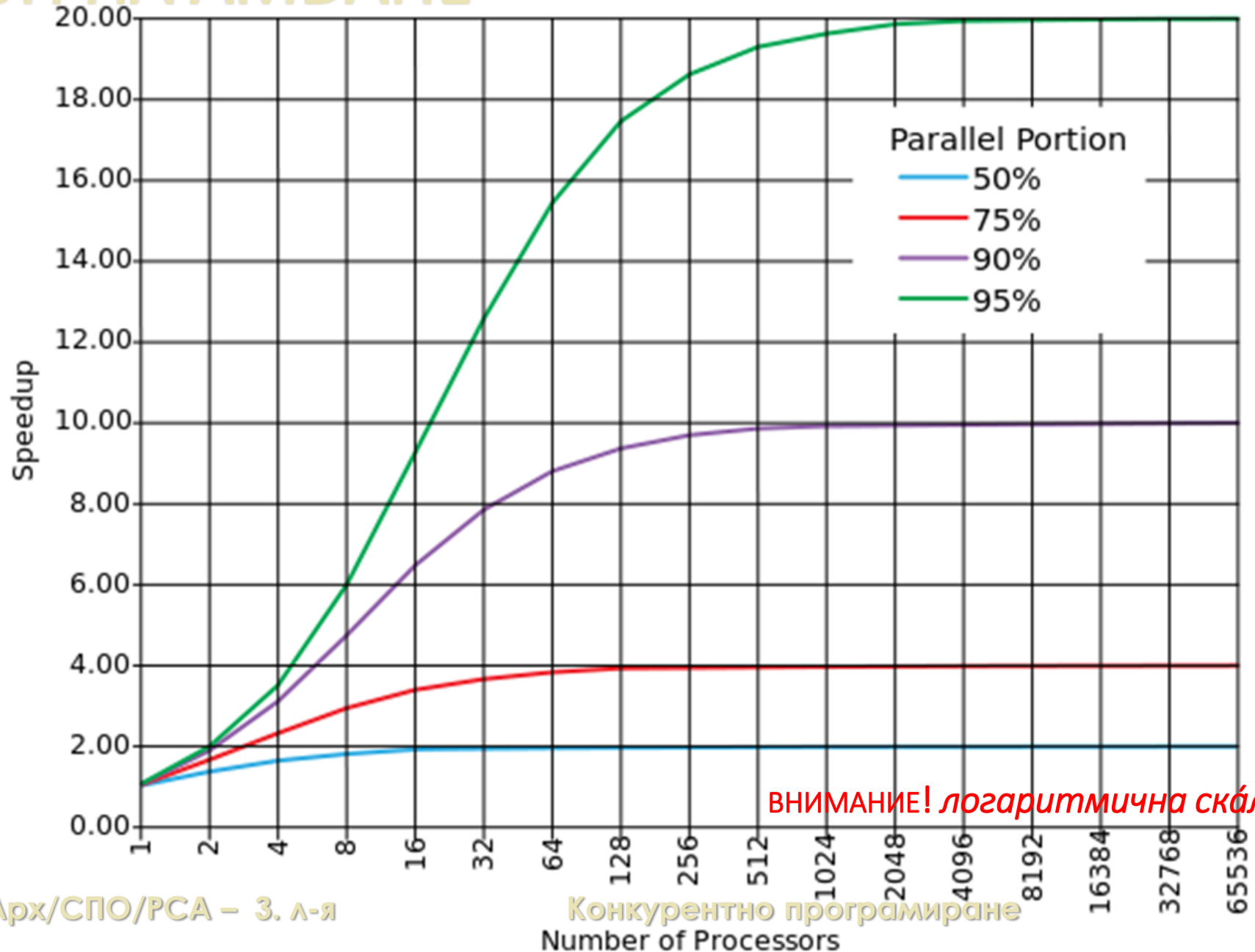
- СЛОЖНОСТТА НА ПОСЛЕДОВАТЕЛНИТЕ АЛГОРИТМИ (БРОЙ ОПЕРАЦИИ) СЕ ОЦЕНЯВА КАТО ФУНКЦИЯ САМО НА РАЗМЕРА НА ПРОБЛЕМНАТА ОБЛАСТ И СЛЕДОВАТЕЛНО МОЖЕ ДА СЕ ОЦЕНИ АБСТРАКТНО ОТ АРХИТЕКТУРАТА; ПРИ ПА ТЯ Е ФУНКЦИЯ НА АРХИТЕКТУРАТА И НА СРЕДАТА ЗА ПАРАЛЕЛНА ОБРАБОТКА (ОСОБЕНО ПРИ ДИНАМИЧНО ПЛАНИРАНЕ)
- ОСНОВЕН ФАКТОР ПРИ ПА Е СТЕПЕНТА НА **ПАРАЛЕЛИЗЪМ**  $P$  – МАКСИМАЛНИЯ БРОЙ ОПЕРАЦИИ, КОИТО МОГАТ ДА СЕ ИЗПЪЛНЯТ ПЪРЯКЕЛНО ПРИ ОБРАБОТКАТА НА АЛГОРИТЪМА – ТОВА Е АРХИТЕКТУРНО НЕЗАВИСИМА ВЕЛИЧИНА; ПРИ РАЗМЕР НА ПРОБЛЕМА  $W$  НЕ ПОВЕЧЕ ОТ  $P(W)$  ПРОЦЕСОРА МОГАТ ДА СЕ ПОЛЗВАТ ЕФЕКТИВНО; СЪЩЕСТВЕНО Е СЪОТНОШЕНИЕТО МЕЖДУ ПАРАЛЕЛНИТЕ И ПОСЛЕДОВАТЕЛНИТЕ СЕГМЕНТИ НА ПА

# ЗАКОН НА АМДАHL (1967):

- ПРИ НАЛИЧИЕ НА ДВЕ ИНТЕНЗИВНОСТИ (ТЕМПОВЕ) НА ОБРАБОТКА НА ДАДЕН ПОРБЛЕМ — ВИСОКО-ПАРАЛЕЛНА  $R_h$  И НИСКО-ПАРАЛЕЛНА  $R_l$ , КОИТО СА В СЪОТНОШЕНИЕ  $f:(1-f)$  ПО БРОЙ НА ГЕНЕРИРАНИ РЕЗУЛТАТИ (МЕЖДИННИ И КРАЙНИ) — ОБЩАТА ИНТЕНЗИВНОСТ НА ОБРАБОТКА Е
- $R(f) = [f/R_h + 1-f)/R_l]^{-1}$
- СЛЕДОВАТЕЛНО ЗА  $f \rightarrow 1$   $R(f) \rightarrow R_h$  И ПРИ  $f \rightarrow 0$   $R(f) \rightarrow R_l$
- Н.В.: МАКАР ЧЕ Е ФОРМУЛИРАН ЗА ТЕМПОВЕ НА ОБРАБОТКА, ЗАКОНА Е В СИЛА И СЕ ПРИЛАГА ЗА АГРЕГИРАНА СТЕПЕН НА ПАРАЛЕЛИЗМА НА ЗАДАНИЕТО

# ...ЗАКОН НА AMDAHL

Amdahl's Law





# УСКОРЕНИЕ И ЕФЕКТИВНОСТ

- ПРИ ОЦЕНКА ИЛИ ИЗМЕРВАНЕ НА УСКОРЕНИЕТО ( $S_p = T_1/T_p$ ) СЕ ПРИЕМА, ЧЕ ВСИЧКИ ПРОЦЕСОРИ В ДВАТА СЛУЧАЯ СА С ИДЕНТИЧНА ПРОИЗВОДИТЕЛНОСТ; ПОРАДИ НАЛИЧИЕ НА КОМУНИКАЦИОННИ И СИНХРОНИЗАЦИОННИ ЗАКЪСНЕНИЯ  $1 < S_p < p$
- **СУПЕРЛИНЕЙНА АНОМАЛИЯ:**  $S_p > p$  МОЖЕ ДА СЕ НАБЛЮДАВА ПРИ НЕОПТИМАЛЕН ПОСЛЕДОВАТЕЛЕН АЛГОРИТЪМ ИЛИ ПРИ ОСОБЕНИ ХАРАКТЕРИСТИКИ НА ПРОБЛЕМА, КОИТО ИЗЯВЯВАТ НИСЪК КАПАЦИТЕТ НА ИЗПОЛЗВАНИЯ ХАРДУЕР: НАПР. ПРИ ГОЛЯМ РАЗМЕР НА ДАННИТЕ (НАДВИШАВАЩ КАПАЦИТЕТА НА ОП) Е ВЪЗМОЖНО ЗНАЧИТЕЛНО ЗАКЪСНЕНИЕ НА ПОСЛЕДОВАТЕЛНАТА ОБРАБОТКА НА ПРОБЛЕМА ПОРАДИ БАВНИ ОПЕРАЦИИ С ВЪНШНАТА ПАМЕТ, ДОКАТО ПРИ ПАРАЛЕЛНА ОБРАБОТКА С РАЗДЕЛЯНЕТО НА ДАННИТЕ МЕЖДУ ВЪЗЛИТЕ ТОЗИ ПРОБЛЕМ ОТПАДА (ОПТИМИЗИРАНЕ НА ПОСЛЕДОВАТЕЛНИЯ АЛГОРИТЪМ В ТОЗИ СЛУЧАЙ НЕ Е ВЪЗМОЖНО КОГАТО ЗА ДАННИТЕ НЕ ВАЖИ ПРИНЦИПА НА ЛОКАЛНОСТ — НАПР. ПРИ МНОГО ПРОБЛЕМИ ОТ AI)
- **ЗАДАЧА:** ДА СЕ АНАЛИЗИРА УСКОРЕНИЕТО ЗА СУМИРАНЕ НА  $N$  СЪБИРАЕМИ В  $P$ -ПРОЦЕСЕН ХИПЕРКУБ, КАТО СЕ ПРИЕМЕ, ЧЕ ВРЕМЕТО ЗА 1 СУМИРАНЕ И ЗА 1 ОБМЕН НА РЕЗУЛТАТ МЕЖДУ ПРОЦЕСИТЕ Е 1. (ВЖ. ФИГ. 3.17)
- **НЕМОНОТОННА АНОМАЛИЯ:**  $\exists p > p_{\text{sat}}: S_p > S_{p+1}$
- ЕФЕКТИВНОСТТА, КОЯТО Е НОРМАЛИЗИРАНО УСКОРЕНИЕ ( $E_p = S_p/p = T_1/(pT_p) < 100\%$ ), ХАРАКТЕРИЗИРА ЧАСТТА ОТ ОБЩОТО ВРЕМЕ ЗА ПАРАЛЕЛНА ОБРАБОТКА, ПРЕЗ КОЯТО ПРОЦЕСОРНИТЕ ЕЛЕМЕНТИ СЕ ИЗПОЛЗВАТ ЕФЕКТИВНО

# ЦЕНА И КОЕФИЦИЕНТ НА ИЗПОЛЗВАНЕ

- ЦЕНА (COST) ПРИ ОБРАБОТКАТА НА ПА С  $p$  ПРОЦЕСОРА ЗА  $T_p$  ЕДИНИЦИ ВРЕМЕ (Н.В. ЕДИНИЦА ВРЕМЕ Е ВРЕМЕТО ЗА ИЗПЪЛНЕНИЕ НА ЕДНА ЕЛЕМЕНТАРНА ОПЕРАЦИЯ) Е
- $C_p = pT_p$
- Т.Е.  $C_p$  Е КРИТЕРИЙ ЗА МАКСИМАЛНИЯ БРОЙ ОПЕРАЦИИ, КОИТО БИХА МОГЛИ ДА СЕ ИЗВЪРШАТ ЗА ВРЕМЕТО НА ОБРАБОТКА НА СЪОТВЕТНИЯ ПА
- КОЕФИЦИЕНТ НА ИЗПОЛЗВАНЕ (UTILIZATION) ПРИ ОБРАБОТКАТА НА ПА, СЪСТОЯЩ СЕ ОТ  $O_p$  НА БРОЙ ОПЕРАЦИИ С  $p$  ПРОЦЕСОРА Е
- $U_p = O_p/C_p = O_p/(pT_p)$
- Т.Е.  $U_p$  Е ОТНОШЕНИЕТО НА ДЕЙСТВИТЕЛНИТЕ КЪМ ПОТЕНЦИАЛНИТЕ ОПЕРАЦИИ ПРИ ОБРАБОТКА НА СЪОТВЕТНИЯ ПА

# ПРОБЛЕМИ ПРИ МНОГОНИШКОВАТА ОБРАБОТКА С МУЛТИПРОЦЕСОР

- НИСЪК ПРАГ НА СОФТУЕРНИЯ ПАРАЛЕЛИЗЪМ
  - НИСКА ИЗЧИСЛИТЕЛНА СЛОЖНОСТ
  - СЪЩЕСТВЕНА ПОСЛЕДОВАТЕЛНА ЧАСТ ОТ АЛГОРИТЪМА - СЛЕДСТВИЕ ОТ 3-НА НА АМДАЛ
- ГРАНУЛАРНОСТ НА ЗАДАНИЯТА
  - ЗА ЕФЕКТИВНО (БАЛАНСИРАНО) ПЛАНИРАНЕ
  - ЗА ОПТИМАЛНА РАБОТА НА КЕША L1
- ГРАНУЛАРНОСТ НА ОБМЕНА — ПЛАНИРАНЕ НА ПАМЕТТА И ПРЕМАХВАНЕ НА НЕАДРЕСИРАНИТЕ КОМПОНЕНТИ
- СТАТИЧНО ИЛИ ДИНАМИЧНО БАЛАНСИРАНЕ
- УПРАВЛЕНИЕ НА КЕША ВКЛ. ЛОКАЛИЗАЦИЯ НА ДАННИТЕ ПРИ СИНХРОНЕН АЛГОРИТЪМ (ПРИ ЗАВИСИМОСТИ ПО ДАННИ)
- АДАПТИВНОСТ В РАЗВОЙНА, ТЕСТОВА И ПРОДУКЦИОННА СРЕДА.

# ПРОБЛЕМ: НИСЪК ПРАГ НА СОФТУЕРНИЯ

## ПАРАЛЕЛИЗЪМ

- МАЛКА ИЗЧИСЛИТЕЛНА СЛОЖНОСТ  $C$  НА КОДА ЗА „ПАРАЛЕЛИЗИРАНЕ“ — ЧИСЛОВАТА ОБРАБОТКА НА ПАРАЛЕЛНАТА ЧАСТ ОТ АЛГОРИТЪМА ТРЯБВА ДА БЪДЕ НА ПОРЯДЪК ПО-ПРОДЪЛЖИТЕЛНА ОТ СИСТЕМНИЯ СВРЪХТОВАР ПО „ПАРАЛЕЛИЗИРАНЕТО“, ВКЛЮЧВАЩ:
  - ПУСКАНЕ НА НИШКИТЕ
  - РАЗПРЕДЕЛЯНЕ НА КОНТЕКСТА НА ЗАДАНИЯТА (ВХОДНИТЕ ДАННИ) И СЛИВАНЕТО НА РЕЗУЛТАТИТЕ (ФАЗА РАЗШИРЯВАНЕ И ФАЗА СВИВАНЕ)
  - СИНХРОНИЗАЦИЯТА (ОБМЕНА) МЕЖДУ НИШКИТЕ (АКО ИМА)
- ИЗВЪН ОБХВАТА НА ПАРАЛЕЛИЗМА СА ВСИЧКИ
  - МРЕЖОВИ ОБМЕНИ С ДРУГИ ВЪЗЛИ
  - АДРЕСИРАНЕ НА СЕРИЙНИ УСТРОЙСТВА (ФАЙЛОВЕ — ВКЛ. ДИСК, ГРАФИКА) ИЛИ АРБИТРИРАНИ УСТРОЙСТВА (L2 КЕШ)

# ПРОБЛЕМИ: ГРАНУЛАРНОСТ

- ГРАНУЛАРНОСТ НА ЗАДАНИЯТА при SPMD  $g$ :

$g = N/p \geq 1$ ,  $N$  – брой подзадания;  $p$  – брой обработващи нишки;

$g \in [10^0, 10^1]$  – едра;  $g \in [10^1, 10^{1.5}]$  – средна;  $g > 10^{1.5}$  – фина;

- никога не работим с фина грануларност
- едрата грануларност е допустима **само при** гарантирано балансиран товар (балансирана матрица)
- грануларността се оптимизира с оглед на 2 контролни процеса:
  - балансирано разпределяне на заданията между обработващите нишки (статично и динамично)
  - оптимално управление на кеша
  - **задача**: дори при последователната обработка ( $p=1$ ) декомпозирайте заданието си на подзадания, така че данните да се съберат в L1 и сравнете  $T_1(g)$
- ГРАНУЛАРНОСТ НА ЗАДАНИЯТА при MPSPD (конвейер, pipeline, в PCA pipes-filters шаблон):

по броя фази (stages)

# ПРОБЛЕМИ: «ГРАНУЛАРНОСТ» НА ОБМЕНА (СИНХРОНИЗАЦИЯТА)

- БРОЯ НИШКИ СЕ СЪОБРАЗЯВА С
  - МАШИНИЯ ПАРАЛЕЛИЗЪМ
  - СЪСТЕЗАТЕЛНИЯ ДОСТЪП МЕЖДУ ТЯХ — ЗАВИСИ ОТ
    - ТОПОЛОГИЯТА НА ЗАВИСИМОСТИТЕ: СРАВНЕТЕ ОБЩА (ГЛОБАЛНА) ПРОМЕНЛИВА МЕЖДУ
      - 2 НИШКИ (ОБИКНОВЕНО ПРИ КОНВЕЙЕР — ТОПОЛОГИЯ «ЛИНИЯ» ИЛИ «ПРЪСТЕН») И
      - ВСИЧКИ НИШКИ — ТОПОЛОГИЯ «ЗВЕЗДА»
    - ПЕРИОДИЧНОСТТА НА ОБМЕН: ТОПОЛОГИЯ «ЗВЕЗДА» НЯМА ДА Е “ОПАСНА” АКО ПЕРИОДИТЕ НА АСИНХРОННА ОБРАБОТКА НА ВСЯКА НИШКА СА ПРОДЪЛЖИТЕЛНИ И ТО С ВРЕМЕВА ДИСПЕРСИЯ (РАЗМЕСТЕНИ ВЪВ ВРЕМЕТО) - ТОВА СЕ ПОСТИГА ПРИ ПО-ЕДРА ГРАНУЛАРНОСТ И СИЛНО СЕ ВЛОШАВА ПРИ ПО-ФИНА ГРАНУЛАРНОСТ

# ПРОБЛЕМИ: СТАТИЧНО ИЛИ ДИНАМИЧНО БАЛАНСИРАНЕ

- СТАТИЧНОТО БАЛАНСИРАНЕ НЕ ПОРАЖДА СВРЪХТОВАР И СЕ СВЕЖДА ДО ИЗБОР НА ОПТИМАЛНА ГРАНУЛАРНОСТ НА ЗАДАНИЯТА
- ДИНАМИЧНОТО БАЛАНСИРАНЕ СЕ СЪСТОИ ОТ 3 ПРОЦЕСА — ИНФОРМАЦИОНЕН, ЛОКАЦИОНЕН И РАЗПРЕДЕЛИТЕЛЕН
- ДИНАМИЧНОТО БАЛАНСИРАНЕ Е ЗАБРАНЕНО — ОСВЕН КАТО ЧАСТ ОТ ПРОЕКТ ЗА ОЦЕНКА НА СВРЪХТОВАРА, КОЙТО ПОРАЖДА ТО
  - ЦЕНТРАЛИЗИРАНО ДИНАМИЧНО БАЛАНСИРАНЕ — ТОПОЛОГИЯ «ЗВЕЗДА»: ЛОША СКАЛИРУЕМОСТ
  - РАЗПРЕДЕЛЕНО ДИНАМИЧНО БАЛАНСИРАНЕ — ЗАВИСИ ОТ ТОПОЛОГИЯТА (БРОЯ ВЪЗЛИ-СЪСЕДИ) — «ДИФУЗНИ» СХЕМИ — НАПР. В ЛИНИЯ (ВЖ. ДЪСКА В КРАЯ)

# ПРОБЛЕМИ: УПРАВЛЕНИЕ НА КЕША

- РАЗМЕРА НА СЕГМЕНТА ДАННИ В L1-КЕША ЗА ДАДЕНА НИШКА ЗАВИСИ ОТ ФОНОВОТО НАТОВАРВАНЕ (BACKGROUND LOAD)
- L1-СЕГМЕНТА НА НИШКАТА СЕ СИНХРОНИЗИРА В ЗАВИСИМОСТ ОТ ОПЕРАЦИИТЕ ЧЕТЕНЕ ИЛИ ЗАПИС (ПРИ ЧЕТЕНЕ СИНХРОНИЗАЦИЯТА СЕ ОТЛАГА И Е РЯДКА)
- СИНХРОНИЗАЦИЯТА МЕЖДУ L1 И L2 СЕ ИЗВЪРШВА НА БЛОКОВЕ („CACHE LINES“) С ФИКСИРАН РАЗМЕР, КОЙТО Е ДИНАМИЧЕН НАСТРОЙВАЕМ ПАРАМЕТЪР ТИПИЧНО ОТ 64-128-256 БАЙТА
- ПРИМЕР: ПРИ  $P > P_{\text{THRESHOLD}}$  НЕ ИЗПОЛЗВАЙ ОБЩА ПРОМЕНЛИВА МЕЖДУ ВСИЧКИ НИШКИ — ПОДОБРЕ ПАРЦИАЛНИТЕ РЕЗУЛТАТИ ДА СЕ ПОЛУЧАТ В ЛОКАЛНАТА ПАМЕТ НА ВСЯКА НИШКА И СЛЕД ТОВА ДА СЕ ОБРАБОТЯТ ПОСЛЕДОВАТЕЛНО В  $O(P)$  НА БРОЙ ОПЕРАЦИИ
  - **ЗАДАЧА:** ОЦЕНИ  $P_{\text{THRESHOLD}}$  ПРИ НАМИРАНЕ НА СУМАТА НА ЕЛЕМЕНТИТЕ НА МАСИВ ОТ  $P$  НИШКИ



# ПРОБЛЕМИ: АДАПТИВНОСТ В РАЗВОЙНА, ТЕСТОВА И ПРОДУКЦИОННА СРЕДА

- СОФТУЕРНИЯ ПРОЕКТ ПРЕМИНАВА ПРЕЗ ФАЗИТЕ
  - [АНАЛИЗ]
  - ПРОЕКТИРАНЕ,
  - ТЕСТВАНЕ,
  - РАЗГРЪЩАНЕ.
- СЪОТВЕТНО СРЕДИТЕ ЗА ИЗПЪЛНЕНИЕ НА СЪОТВЕТНАТА ФАЗА СА
  - РАЗВОЙНА (DEVELOPMENT)
  - ТЕСТОВА
  - ПРОДУКЦИОННА (PRODUCTION)
- ЗА QOS ТЕСТОВИЯТ ПЛАН ТРЯБВА ДА СЕ ДОПЪЛНИ СЪС СЛУЧАИ, КОИТО СЪОТВЕТСТВАТ НА ПРОДУКЦИОННАТА СРЕДА (ИНФРАСТРУКТУРА).

# ЕТАЛОННИ ПАРАЛЕЛНИ АЛГОРИТМИ (SPMD)

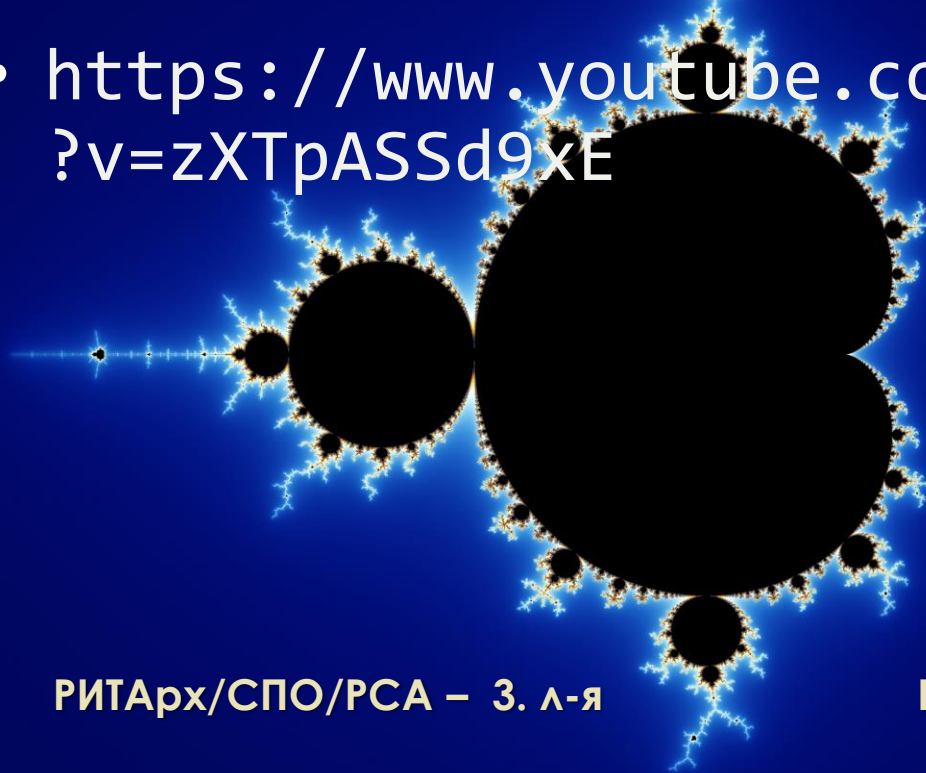
- **АСИНХРОННИ SPMD:** ТЕСТ НА МАНДЕЛБРОТ И ФРАКТАЛНИ (СЕБЕПОДОБНИ) ИЗЧИСЛЕНИЯ:

- $z_{n+1} = z_n^2 + C$ , ЗА  $z_0 = C$

- <http://mathworld.wolfram.com/MandelbrotSet.html>

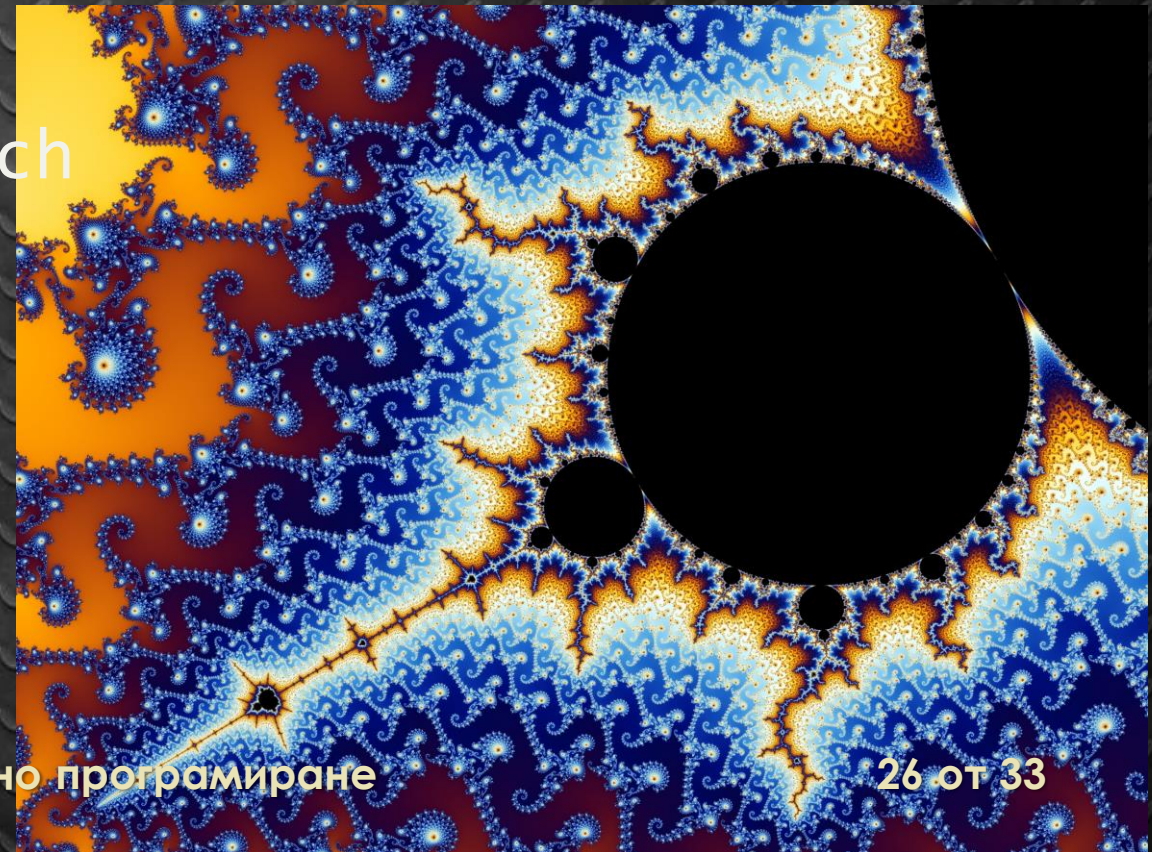
- <https://www.youtube.com/watch?v=zXTrASSd9xE>

• **ЗАДАЧА:** ПОБИТОВО (VETESCODE) КОДИРАНЕ НА СЪСТОЯНИЕТО НА ПИКСЕЛИТЕ – 1 БАЙТ КОДИРА 1 ПИКСЕЛ С 256-ЦВЕТНА ПАЛИТРА. **СРАВНЕТЕ** ЧРЕЗ УСКОРЕНИЕТО **СИСТЕМНИЯ СВРЪХТОВАР (OVERLOAD)** ЗА L1-КЕША ПРИ ПОБИТОВО КОДИРАНЕ И ПРИ КОДИРАНЕ С ЦЯЛА ПРОЦЕСОРНА ДУМА (4 БАЙТА) НА ПИКСЕЛ.



РИТАрх/СПО/РСА – 3. л-я

Конкурентно програмиране

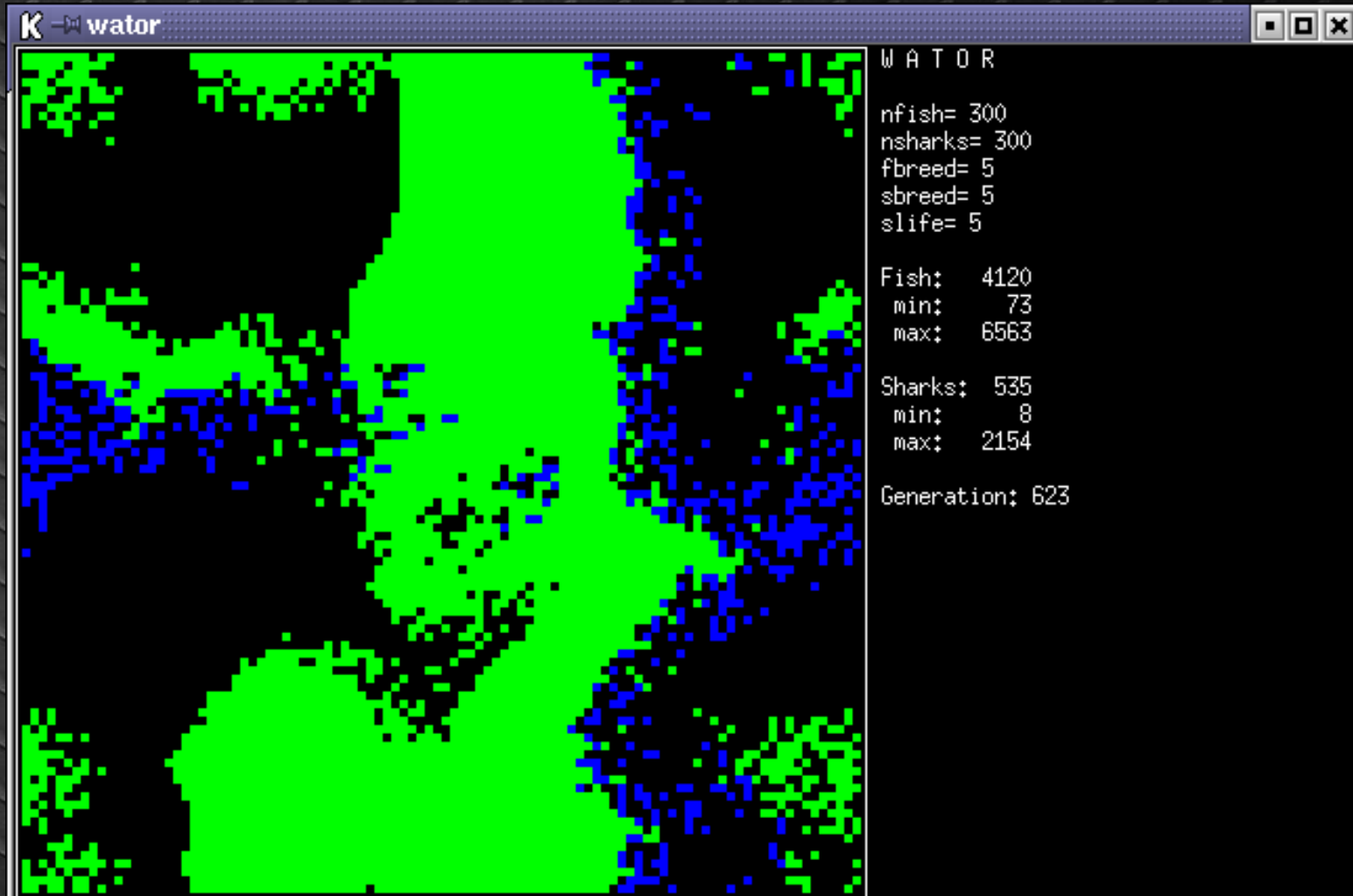


26 от 33

# ЕТАЛОННИ ПАРАЛЕЛНИ АЛГОРИТМИ (SPMD)

- **ЛОКАЛНО-СИНХРОННИ** SPMD: СИМУЛАЦИЯТА WATOR (ИГРИ РАЖДАНЕ-СМЪРТ, ГЕНЕТИЧНИ АЛГОРИТМИ)

- **ЗАДАЧА:** ПОБИТОВО КОДИРАНЕ НА СЪСТОЯНИЕТО НА ПОЗИЦИИТЕ – 1 БАЙТ КОДИРА 4 ПОЗИЦИИ С ПО 2 БИТА ЗА ОЦВЕТАВАНЕТО НА ВСЯКА ПОЗИЦИЯ. **СРАВНЕТЕ** ЧРЕЗ УСКОРЕНИЕТО **СИСТЕМНИЯ СВРЪХТОВАР (OVERLOAD)** ЗА L1-КЕША ПРИ ПОБИТОВО КОДИРАНЕ И ПРИ КОДИРАНЕ С ЦЯЛА ПРОЦЕСОРНА ДУМА НА ПОЗИЦИЯ.

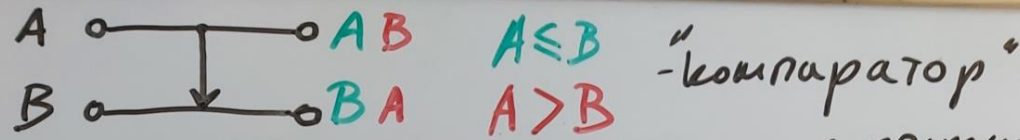


# ЕТАЛОННИ ПАРАЛЕЛНИ АЛГОРИТМИ (SPMD)

## ЛОКАЛНО-СИНХРОННИ SPMD:

### СОРТИРОВКА — СОРТИРАЩИ МРЕЖИ:

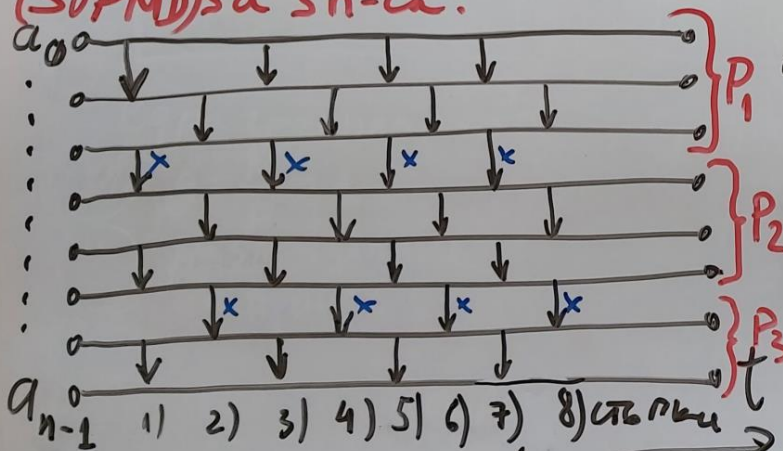
- САМО ODD-EVEN
- ПРИ ПАРАЛЕЛНО „МЕХУРЧЕ“ БАЛАНСЪТ Е НЕВЪЗМОЖЕН



Сортираща мрежа

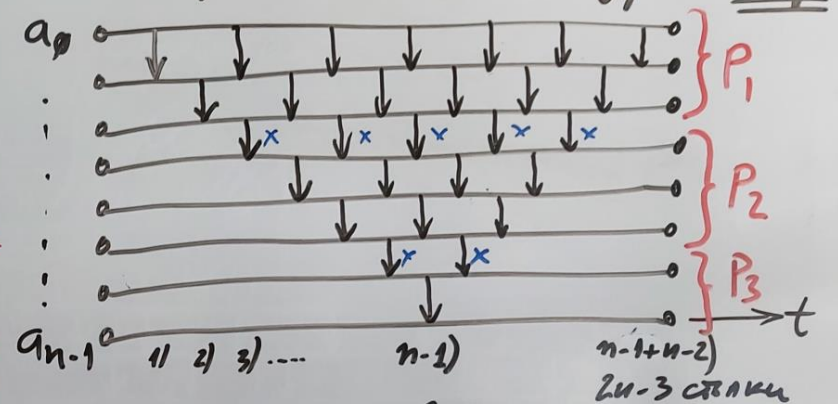
Зелени - незелени;  
 X — синхронизация:

С декомпозиция по данни (SPMD) за 3 п-са:

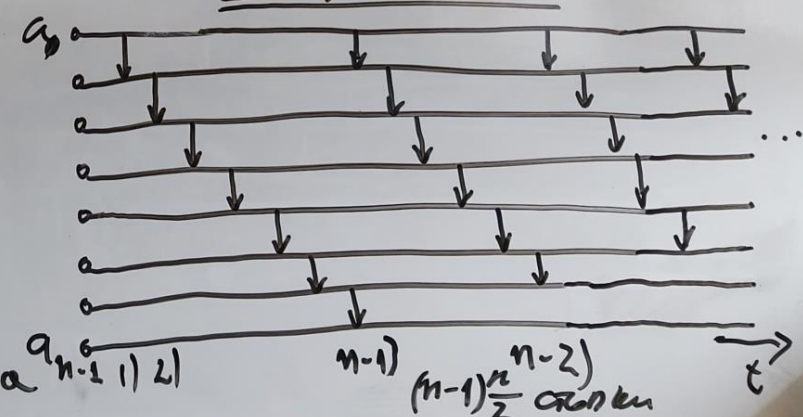


↑ несортирани последователно сортирани  
 (n стълбци) ↑  
 a<sub>n-1</sub> 1) 2) 3) 4) 5) 6) 7) 8) стълбци

сортираща мрежа мехурче-пар.

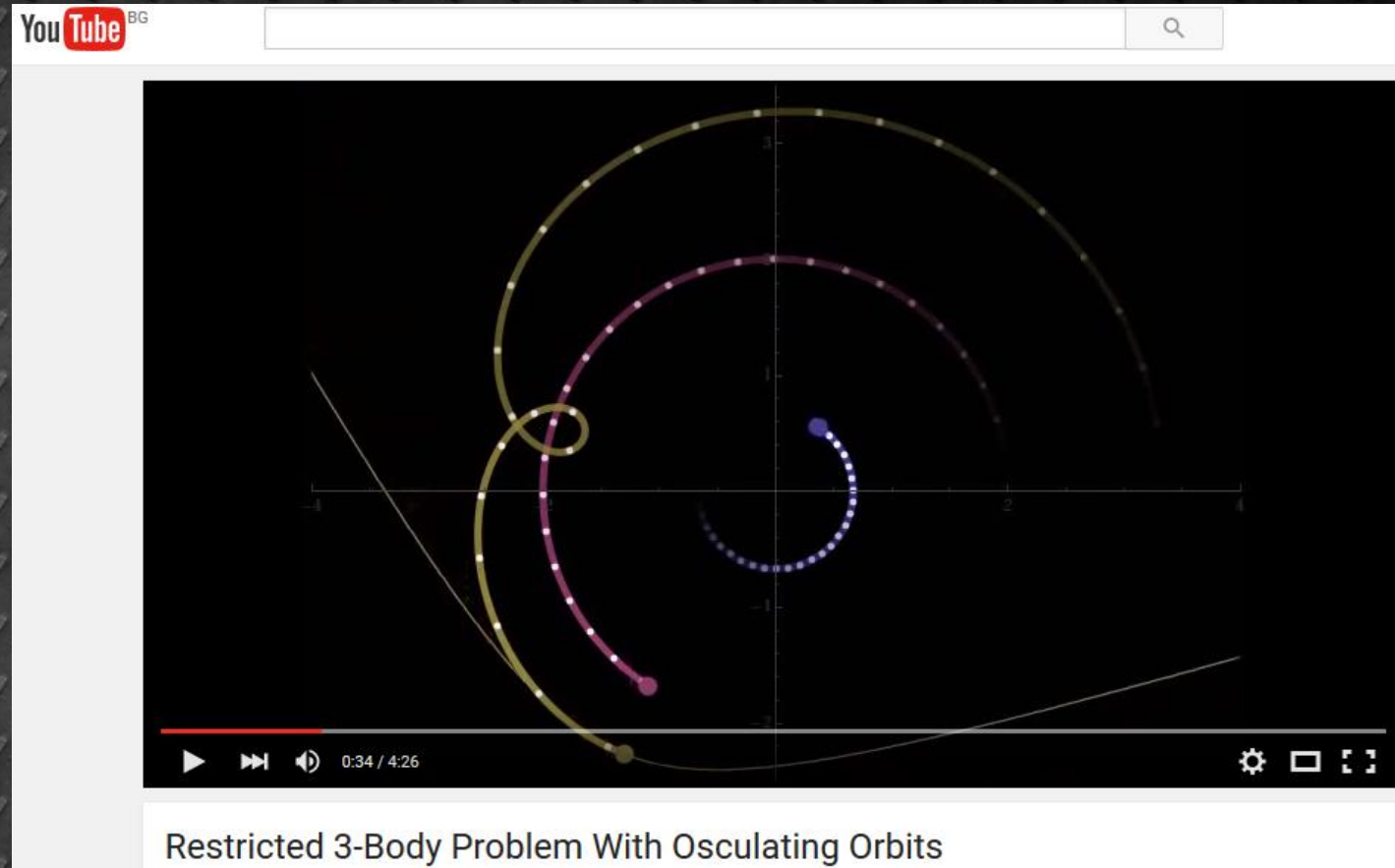


и последователно:

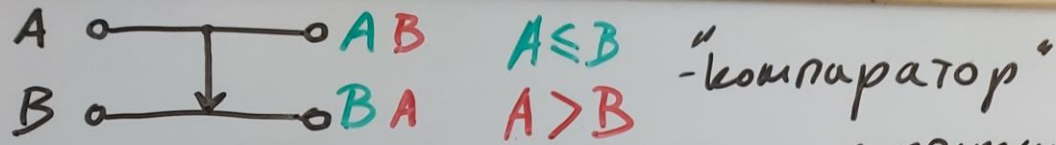


# ЕТАЛОННИ ПАРАЛЕЛНИ АЛГОРИТМИ (SPMD)

- ГЛОБАЛНО-СИНХРОННИ SPMD:  
СИМУЛАЦИЯТА N-BODY
  - [https://en.wikipedia.org/wiki/N-body\\_problem](https://en.wikipedia.org/wiki/N-body_problem)
  - <http://www.cs.cmu.edu/~scandal/alg/nbody.html>



# ПАРАЛЕЛНА СОРТИРОВКА – SPMD

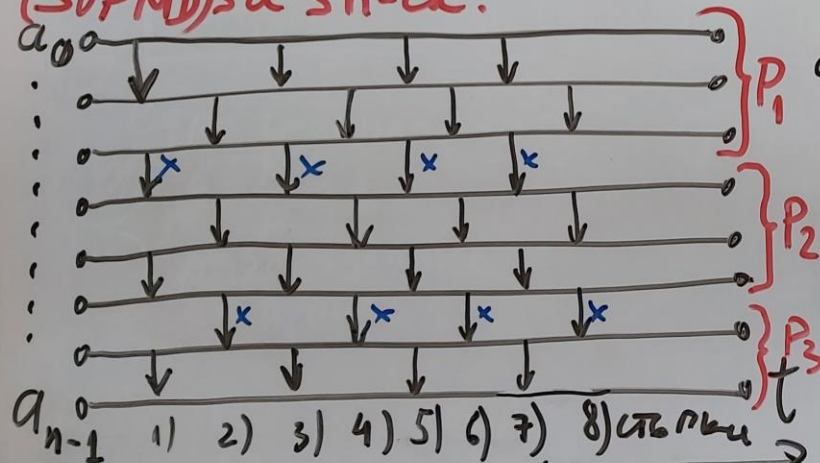


Сортираща мрежа

Забелки - кезебелки:

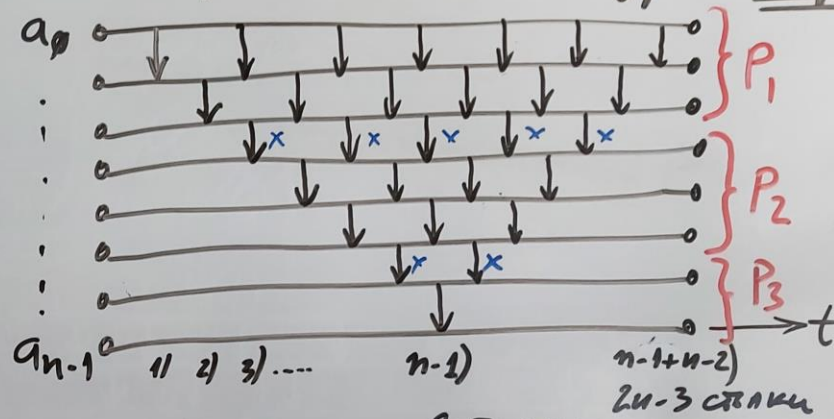
x — синхронизация:

С декомпозиция по данни (SPMD) за 3n-са:

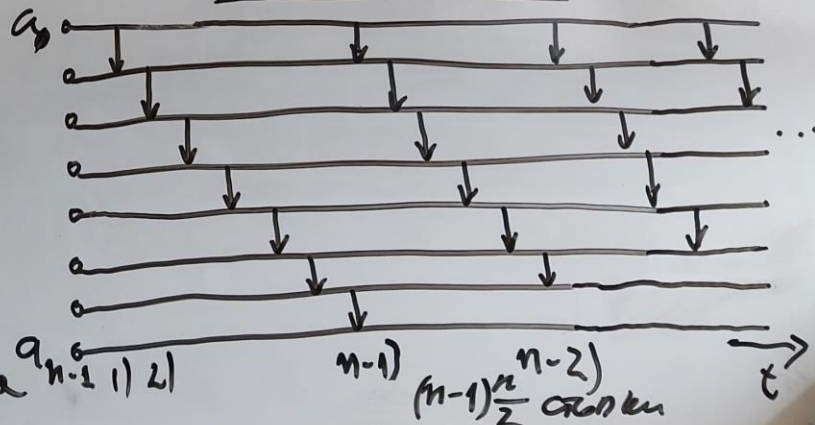


↑ несортирана последователност сортирана

сортираща мрежа мехурче-пар.



и последователно:



• **ЗАДАЧИ:** ОЦЕНЕТЕ (АНАЛИТИЧНО) ИЛИ СРАВНЕТЕ (ЕКСПЕРИМЕНТАЛНО) УСКОРЕНИЕТО НА

1. СОРТИРАНА, АНТИСОТИРАНА И НЕСОТИРАНА ПОСЛЕДОВАТЕЛНОСТ ПРИ МЕХУРЧЕ

2. SPMD-СОТИРАНЕ ПРИ ПАРАЛЕЛНО МЕХУРЧЕ

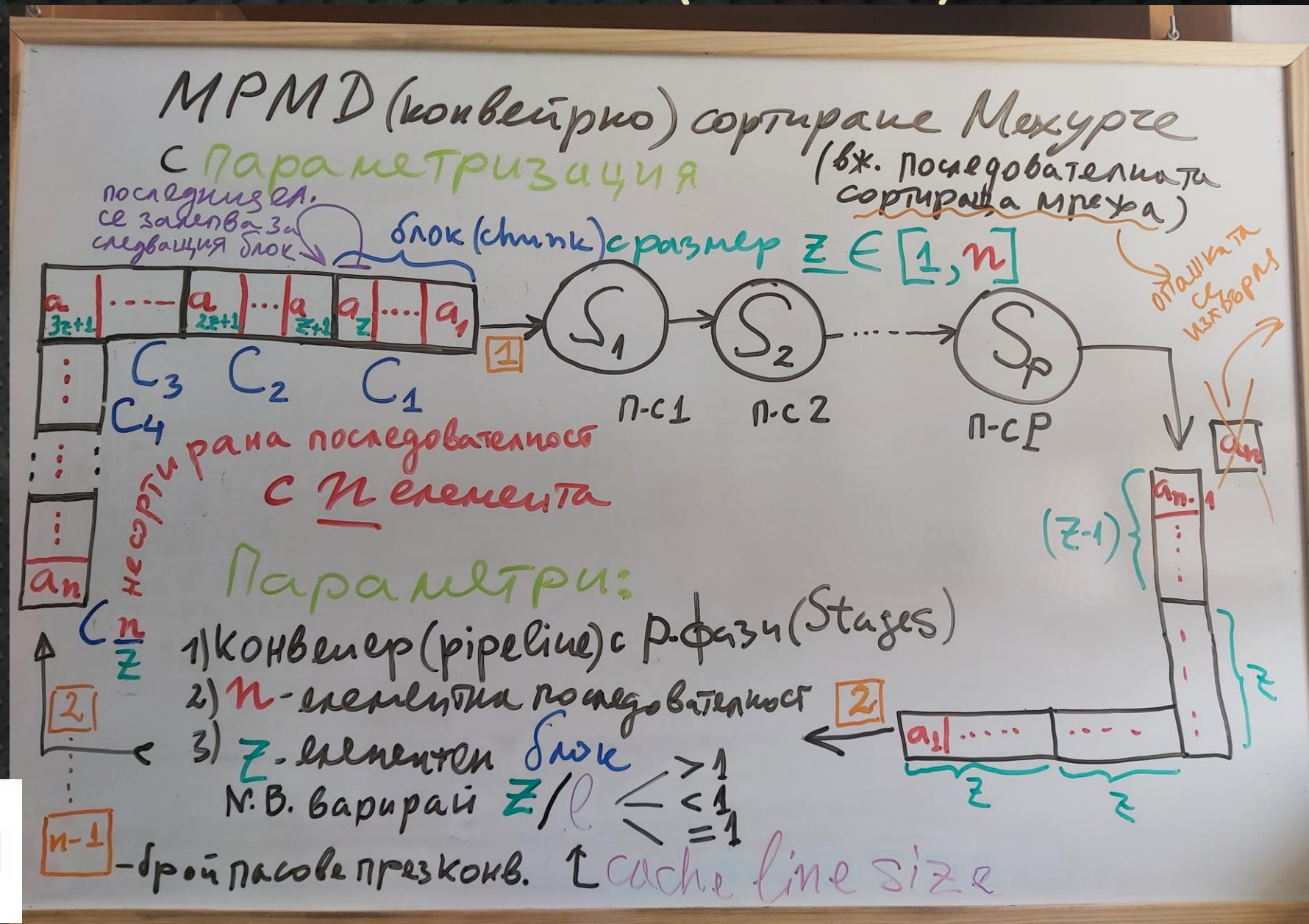
a) с MPMD (НАПР. КОНВЕЙРНО) СОТИРАНЕ ПРИ ПОСЛЕДОВАТЕЛНО МЕХУРЧЕ (ВЖ. 3.31)

b) СЪС SPMD-СОТИРАНЕ ЧЕТНИ-НЕЧЕТНИ

# MPMD-ПАРАЛЕЛИЗЪМ – КОНВЕЙРИ (ЛИНИИ)

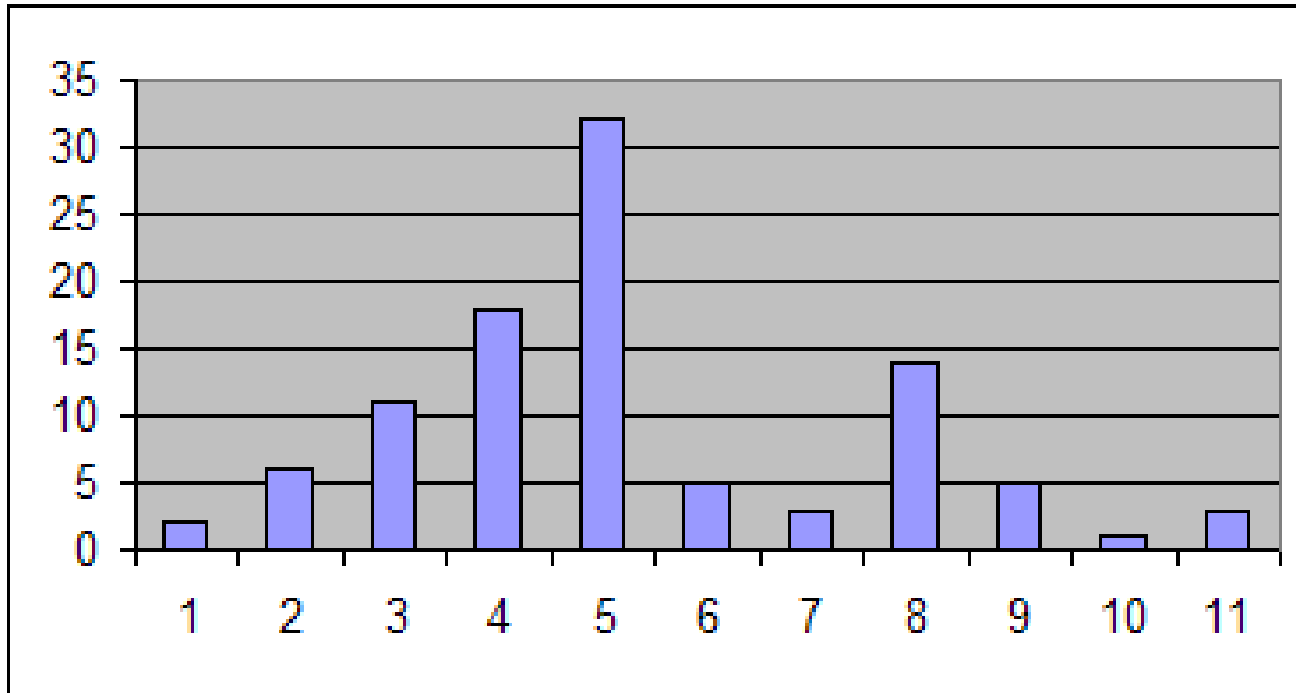
- ЗАДАЧА:** ИЗПЪЛНЕТЕ ПОСЛЕДОВАТЕЛНА СОРТИРОВКА МЕХУРЧЕ НА  $n$  ЕЛЕМЕНТА С  $p$ -ПРОЦЕСЕН КОНВЕЙЕР КАТО ПАРАМЕТРИЗИРАТЕ МОДЕЛ И ИЗСЛЕДВАТЕ ВСИЧКИТЕ МУ ПАРАМЕТРИ

- ЗАДАЧА:** СРАВНЕТЕ УСКОРЕНИЕТО И СЧЕТЕТЕ ХИСТОГРАМА НА БАЛАНСИРАНЕТО ПРИ ИЗЧИСЛЯВАНЕ НА  $e$  ЧРЕЗ ЦИТИРАНИЯ СТЕПЕНЕН РЕД (ЗА  $x=1$ ) СЪС SPMD-ПАРАЛЕЛИЗЪМ И С КОНВЕЙЕР



$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}, \quad x \in (-\infty; +\infty)$$

# ХИСТОГРАМА НА СТОХАСТИЧНИ ПРОЦЕСИ



- ПРИМЕРНА ХИСТОГРАМА НА БРОЯ ИЗПЪЛНЕНИ ЗАДАНИЯ ОТ 11 ПРОЦЕСА ПРИ ДИНАМИЧНО БАЛАНСИРАНЕ
- **ЗАДАЧА:** РЕГИСТРИРАЙТЕ ВРЕМЕТО НА РАБОТА (ИЛИ НА ПРЕСТОЙ) НА ОБРАБОТВАЩИТЕ ПРОЦЕСИ ПРИ ОБЩО ВРЕМЕ НА ОБРАБОТКА И НА ОБРАБОТКАТА  $T_p$  ВЪВЕЖДАЙКИ В ТЕСТА СИ КОМАНДЕН ПАРАМЕТЕР  $V[ERBOSE]$  ЗА РЕГИСТРИРАНЕ НА ВРЕМЕНАТА; ОЦЕНЕТЕ СВРЪХТОВАРА, КОЙТО СЕ ПОРАЖДА ОТ ДОБАВЕНАТА ФУНКЦИЯ НА НАБЛЮДЕНИЕ НА ЛОКАЛНИТЕ ВРЕМЕНА НА ОБРАБОТКА И ПРЕСТОЙ.



# КАРТА НА ЕТАЛОННИТЕ ПРИЛОЖЕНИЯ И ПАРАМЕТРИТЕ ИМ

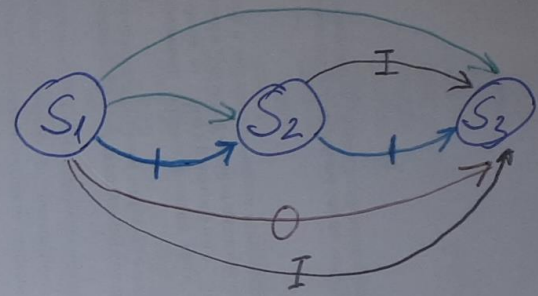
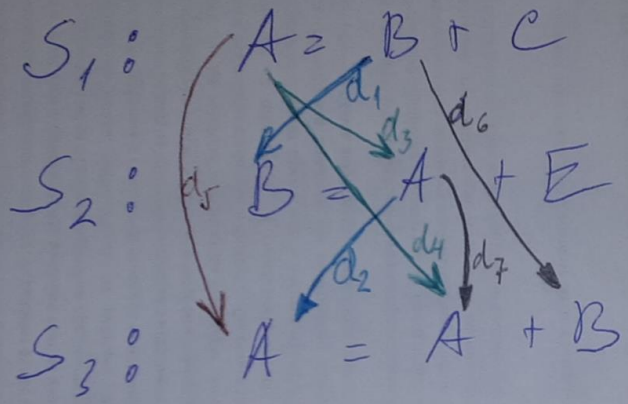
	Код 2 →	A	B	C	D	E	D	F	G
К о д 1 ↓	Параметър →	Изчислителна сложност C	Грануларност $g = N/p$ или $p$ - <i>max</i> за съпоставима грануларност	Адаптивно ст към D-cache и „cache line“ $l$	Модел (топология)		Балансиране		
	Алгоритъм ↓				SPMD (звезда)	MPMD (линия, решетка, хиперкуб...)	Статично	Динамично	
								Централизирано	Разпределено
1	Mandelbrot	X	3	4-5	При 1.F	При 1.G	1	5	6
2	WaTor	5 - 6	?	6	X	1	1	?	?
3	OddEven	3 - 4	?	4 - 5	X	1	1	?	?
4	Bubble	4	5	6	X	1	1	?	?
5	Степенен ред (e)	X	4	5	X	1	1	6	?
6	nBody	5 - 6	X	6	1	?	1	?	?

- Асинхронни, локалносинхронни и глобалносинхронни алгоритми
- X – забранено или невъзможно; 1 – подразбиращо се (задължително); ? – съмнителна опция
- X и ? са допустими като допълващ вариант за съпоставка с основния – тогава носят „точки“

# [MULTITHREADING, “HIPERTHREADING”]

- СЪВМЕСТЯВАНЕ НА ЕДНОВРЕМЕННОТО ОБРАБОТВАНЕ НА 2 ПРОЦЕСА ОТ 1 СУПЕРСКАЛАРНО ЯДРО ПОСРЕДСТВОМ 2 ЛОГИЧЕСКИ ЯДРА
  - СЪМНИТЕЛНО Е, ЧЕ ВИНАГИ ТОВА ВОДИ ДО ПО-ДОБРА ПРОИЗВОДИТЕЛНОСТ (В НИКАКЪВ СЛУЧАЙ НЕ ДВОЙНА!!!) ПРЕДВИД И ДОБАВЕНИЯ СИСТЕМЕН СВРЪХТОВАР
- ИНТЕЛ РЕКЛАМИРА 0-15-30% УСКОРЕНИЕ В ЗАВИСИМОСТ ОТ ОС ПРИ 5% ЗАВИШЕНА КОНСУМАЦИЯ НА ЕЛЕКТРОЕНЕРГИЯ
- ТЕОРЕТИЧНО (В СИЛНА ЗАВИСИМОСТ ОТ ИМПЛЕМЕНТАЦИЯТА НА ОС-ЯДРОТО) СЪВМЕСТЯВАНЕ НА ИНСТРУКЦИИ ОТ КОНТРОЛНИЯ ПОТОК НА 2 ПРОЦЕСА В НЯКОЛКО ( $m$  НА БРОЙ) ИНСТРУКЦИОННИ КОНВЕЙРА НА 1 ПРОЦЕСОРНО ЯДРО – САМО ПРИ НЕЗАВИСИМОСТ НА СЪВМЕСТЕНИТЕ ИНСТРУКЦИИ
- ПО-ПРАКТИЧНО ЗНАЧЕНИЕ ИМА СЪВМЕСТЯВАНЕ НА РАБОТАТА НА СУПЕРСКАЛАРНОТО ЯДРО ПО НЯКОЛКО ПОСЛЕДОВАТЕЛНИ ИНСТРУКЦИИ НА  $P_1$  С ОПЕРАЦИИ ПО СИНХРОНИЗАЦИЯТА МЕЖДУ L1- И L2-КЕША ЗА  $P_2$  (И ТО САМО КОГАТО ЗА  $P_1$  НЕ СЕ ИЗВЪРШВАТ ОПЕРАЦИИ С L1)

3.5-6.



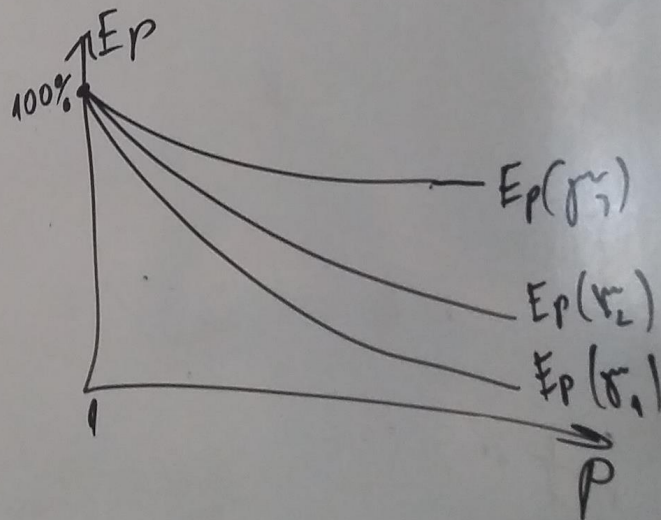
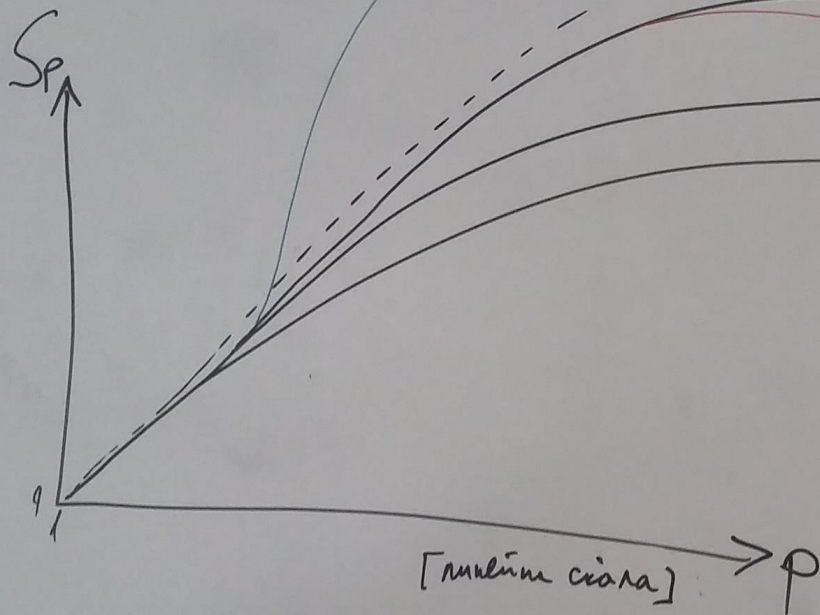
последовательные выражения

граф. на зависимости  
dependency graph

- непреобразимы  $\dashrightarrow$   $\longrightarrow$  по данным
- преобразимы (removable)  $\dashrightarrow$ 
  - $\dashrightarrow$  +  $\longrightarrow$  антизависимость
  - $\dashrightarrow$  0  $\longrightarrow$  по выходу
  - $\dashrightarrow$  I  $\longrightarrow$  по входу
  - $\dashrightarrow$  [  $\dashrightarrow$  X  $\longrightarrow$  ] по управлению, ключевые (control dependency) if-then-else

$$R(f) = \frac{1}{\frac{f}{R_H} + \frac{1-f}{R_e}}$$

3.17.



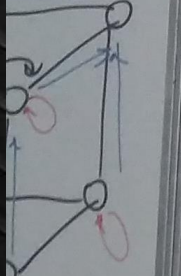
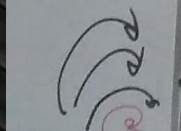
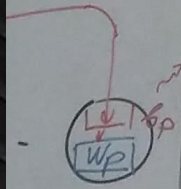
оперативна активність

випередження аналізу

градуальність

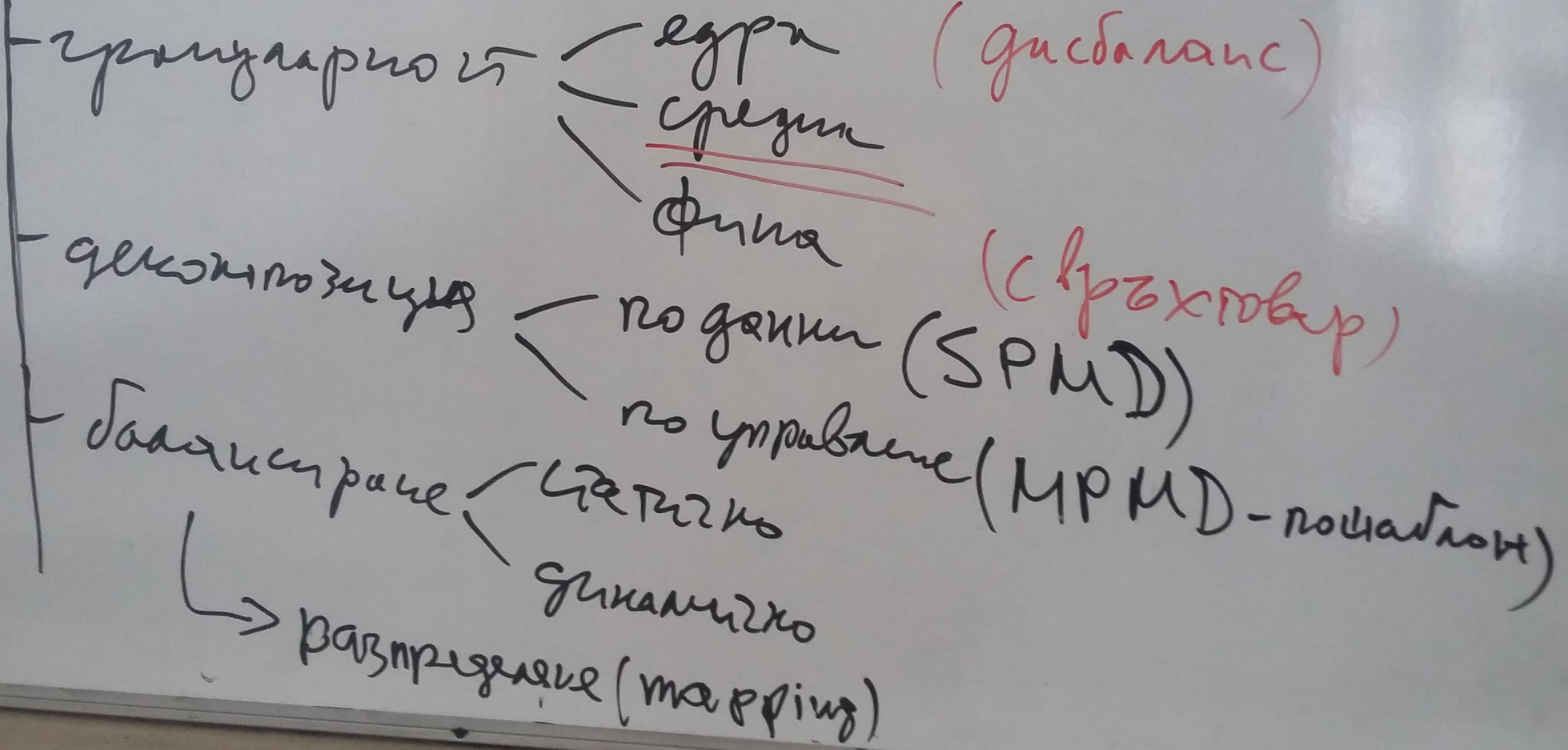
прямий буфер

апродумано  
не в  
линій



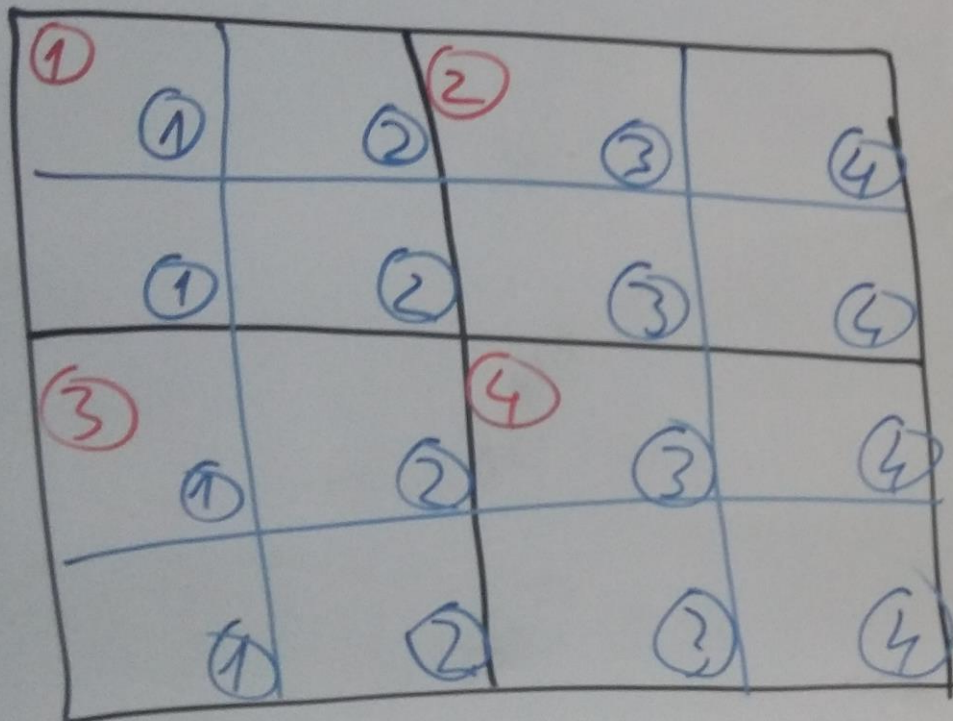
S  
2 lbp  
log<sub>2</sub>

# Паралелен алгоритъм



3.32. —  
сл. 1D

$p=4$  процес  
 $N=$  задачи



Паралелен  
алгоритъм

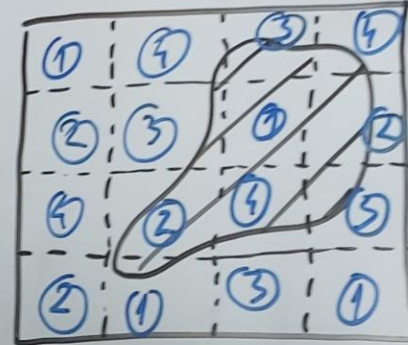
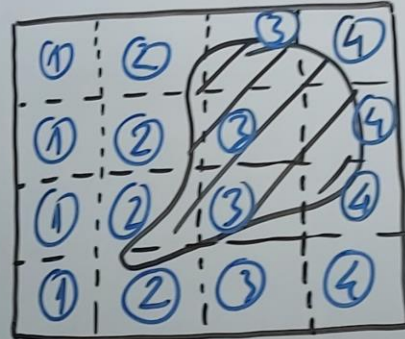
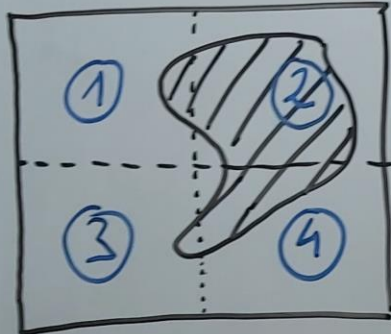
$p=N$  - най-сва трайност

$p \ll N$  - средна трайност

Статично **циклично дисперсно** балансиране  
(scatterd)

3.23

област на теста на М.



едра грануларност  
 $g = \frac{N}{p} = 1$

(i) разпределяне на процес i

зона на висока изт. сложност

средна грануларност  
 $g = \frac{N}{p} = \frac{16}{4} = 4$

с циклично балансиране  
→ проблем: свързати области  
решение: по-фина грануларност

средна грануларност  
 $g = 4$

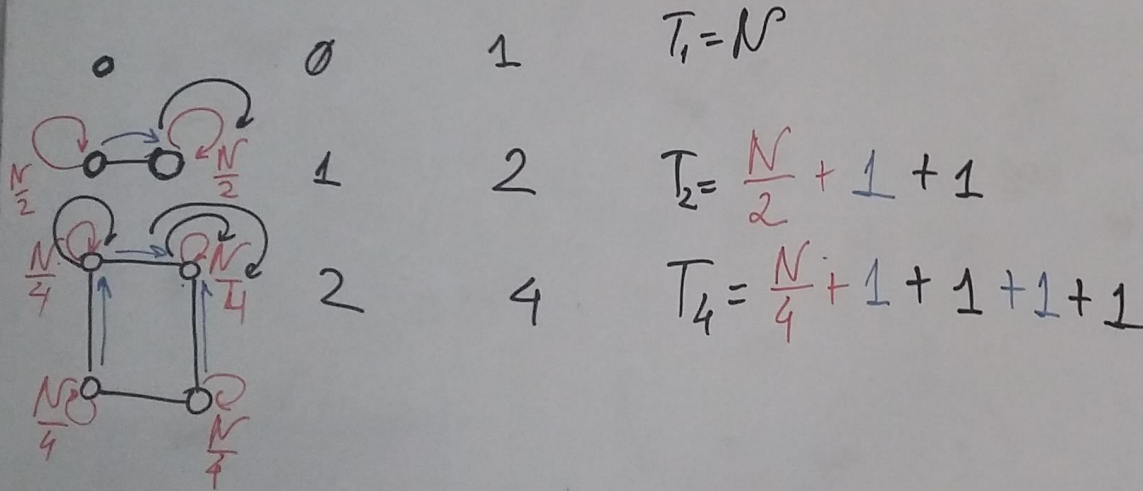
с дисперсно балансиране

$$S_p = ? = \frac{T_1}{T_p} \quad 3.17.$$

свърване на  $N$  свързани  
върху  $p$ -процеси хиперкуб

1 операция свърване = 1 време  
1 операция пообмен = 1 време

Пълна мрежа и комуникация  
Хиперкуб:  $S \rightarrow P \quad T_1 = N$

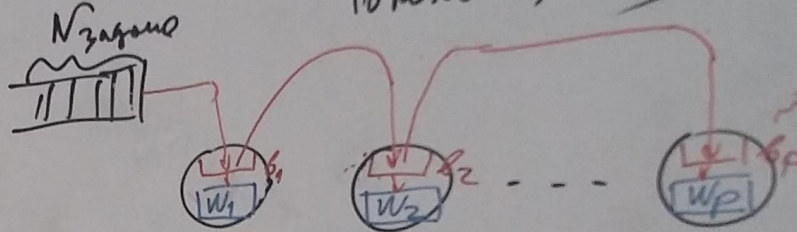


0, 3, 8  $T_8 = \frac{N}{8} + 1 + 1 + 1 + 1 + 1 + 1$

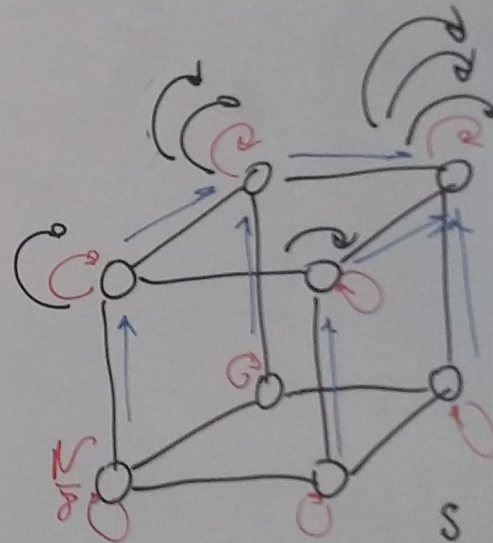
$$T_p = \frac{N}{p} + 2 \underbrace{\log_2 p}_S$$

3.33. —  
сл. 1G

Ефективно разпределение  
на работните процеси в  
тополож. мрежа

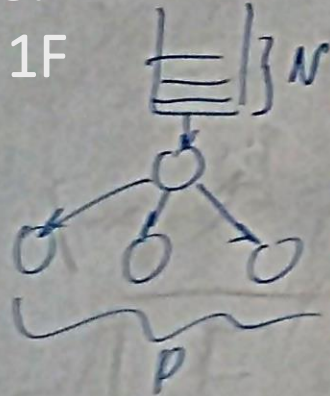


оптимально  
будет





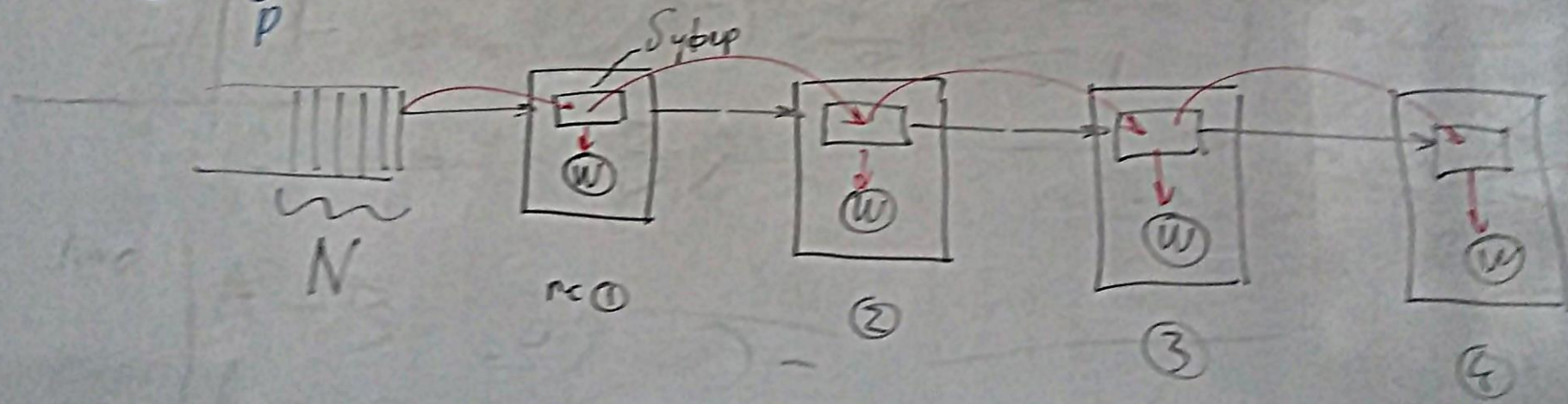
3.33. —  
сл. 1F



динамично балансира  
с централен РС

динамично балансира  
взв. топология "верига"

3.33. —  
сл. 1G



бр. п-си  $p > 1$

# Конкурендно програмиране

паралелно (PRAM)

разпределено (Message Passing)

иерархическа  
обмен му-п-сите

мулти процесор

мулти компютър

обща променлива  
(ключалки)  
lock  
предимно SPMD

EREW  
CREW  
ERCRW  
CRCW

обмен на съобщения  $\rightarrow$  MW  
(middleWare)

предимно MPMD

синхронизация

синхронни  
- локално (съсед)  
- глобално (x:0x)  
(има зависимост)

Watol  
и Body

карт. Client-Server  
+ шаблони от S.B. на  
P2P, конвейер

асинхронни  
нямат зависимост  
(нямат обмен)

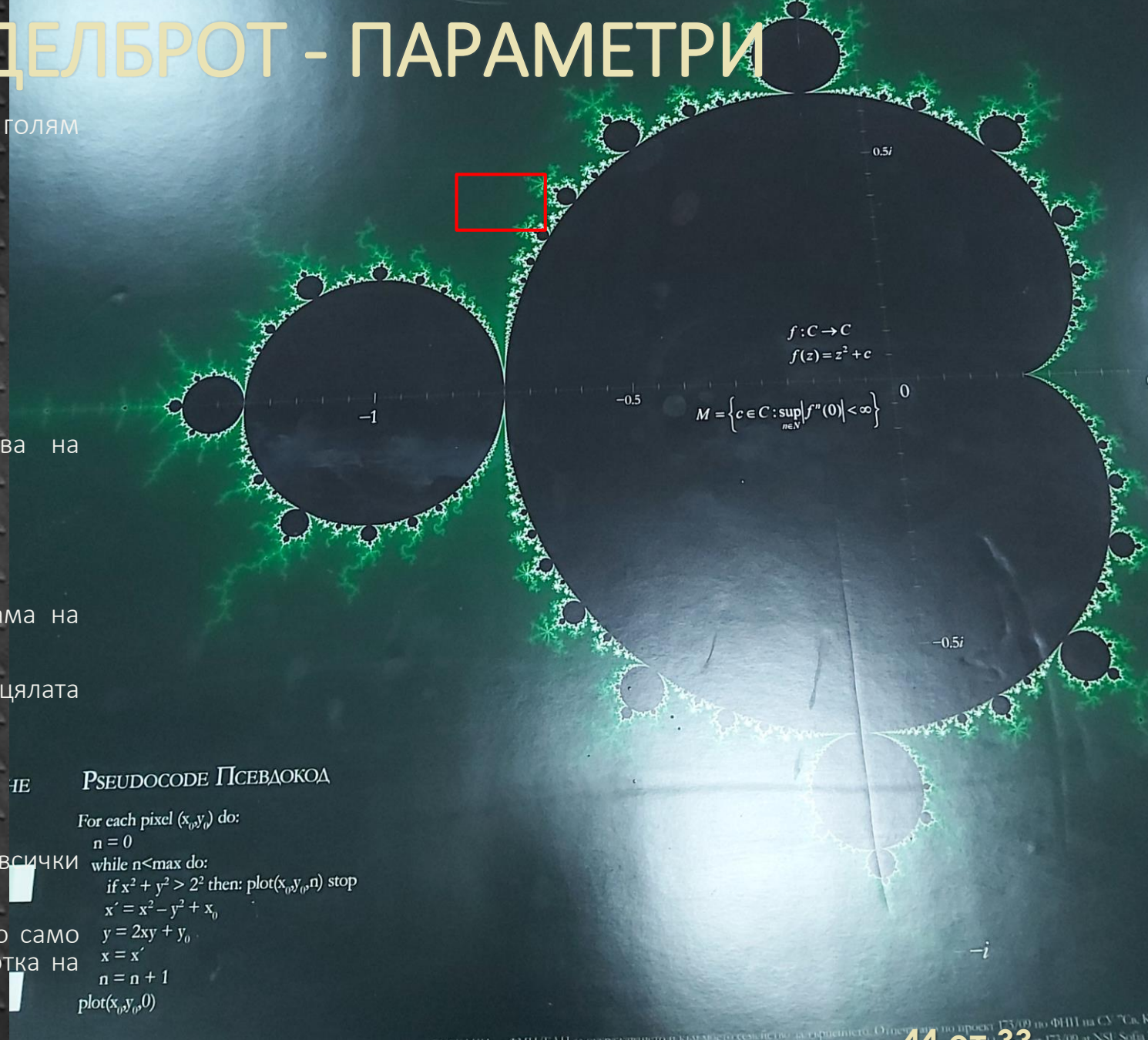
Mandelbrot

палета на обработка (5лж)  
изком конвейер

модел на мултикомпютър					среда
класер	ферма	грид	клауд		3.25.
X					развойна / development
X					тестова + QoS
X	X	X	X		Производство / production

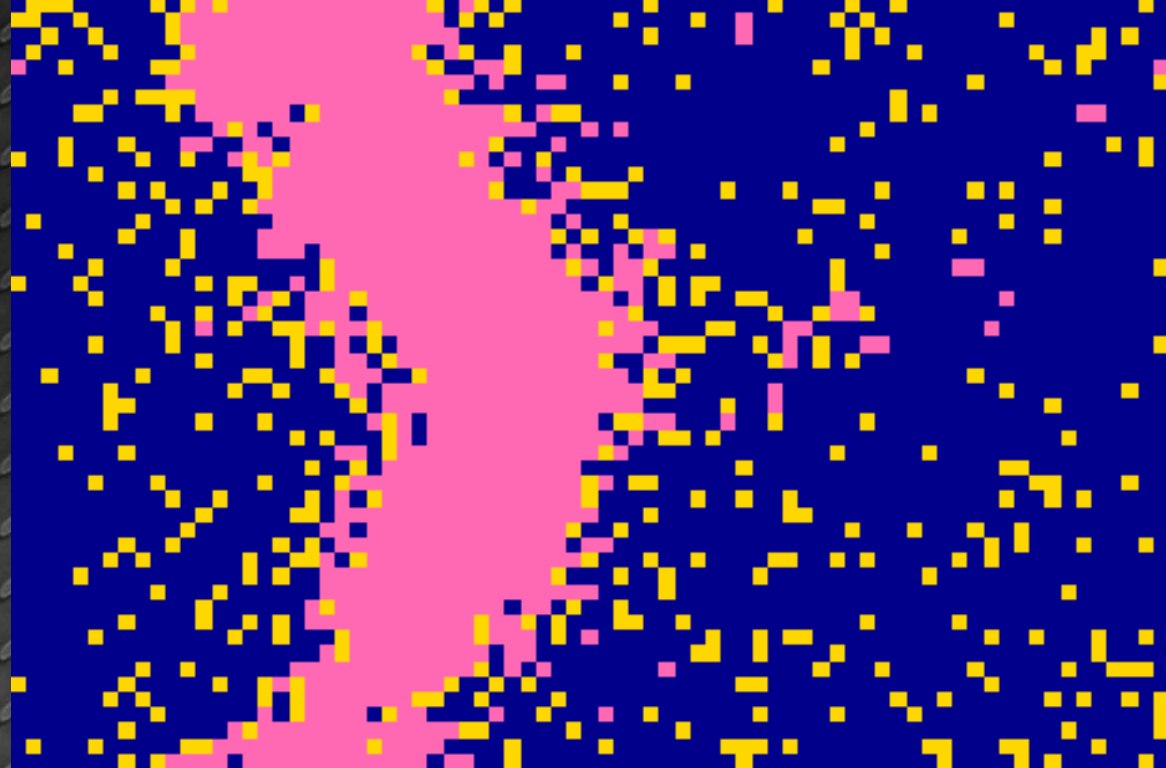
# МНОЖЕСТВО НА МАНДЕЛБРОТ - ПАРАМЕТРИ

- **начална комплексна точка и стъпка** - по избор но с голям дисбаланс
- дълбочина = 256, [1024] цвята
- размер 4K -> 3840 x 2160 pixel
- точка **Функционално проектиране** съдържа:
  - модел на паралелното приложение
  - последователностна диаграма на процесите;
  - опис на командните параметри (който съответства на тестовия план)
- точка **Технологично проектиране** съдържа:
  - функция и код за стартиране на  $p$  нишки
  - функция и код за обмен последователностна диаграма на процесите;
  - функция и код за системно време (начало и край на цялата обработка или на всяка нишка)
- точка **Тестване** включва
  - тестовия план,
  - **XY-диаграми** на семействата криви на ускорението за всички тестови случаи
  - опция: **хистограма на разпределението** на задания, но само при динамично балансиране или по време на обработка на нишките при статично балансиране



# WaTor - ПАРАМЕТРИ

- “видимо” двумерно пространство 2K:1K (координатите по X и по Y са от 0 до 1920/980)
- рибите и празните полета се визуализира с оцветени квадрати 4x4 px
- началната маркировка на океана се генерира предварително рандомизирано в масив с X- и Y-компонента с 1000 херинги и 200 акули (подлежи на корекция за изчислителна сложност C)
- за всяка риба се дефинира **фиксирано** (или като усложнен вариант за C – **рандомизирано**)
- **фиксирано/средно** време на живот 20 стъпки
- **фиксирано/средно** време на гладна смърт 10
- **фиксирано/средно** време на репликиране 5
- херингата нулира брояча за гладна смърт при заемане на нова позиция.
- акулата нулира брояча за гладна смърт при заемане на херинга
- проверката на свободна и хранителна позиция за всяка риба се прави
- фиксирано – в ред север изток юг запад
- търсене на нова позиция – до откриване на възможно или до затваряне на цикъла от че проверки
- приоритета на маркиране на граничните области между процесите е същия – южния процес задава бъдещата маркировка н по линията на северния и западния на източния
- това значи че всеки процес обработва своята приоритетна зона първо и веднага я предава на своя съсед, а след това обработва локалната си област



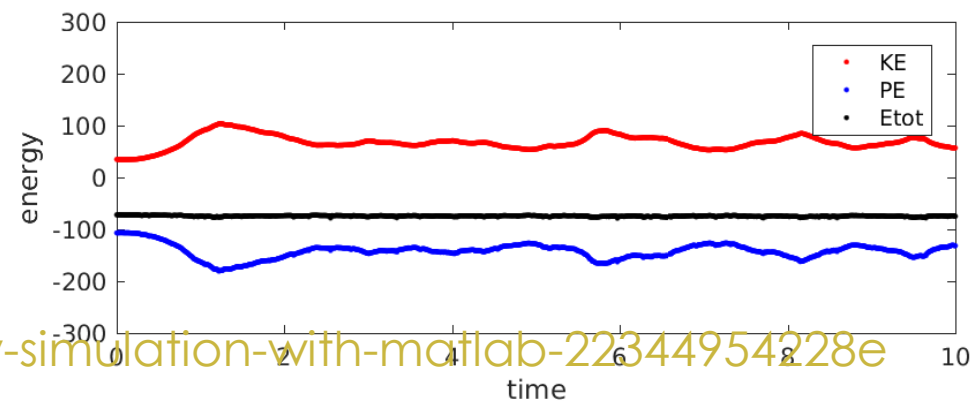
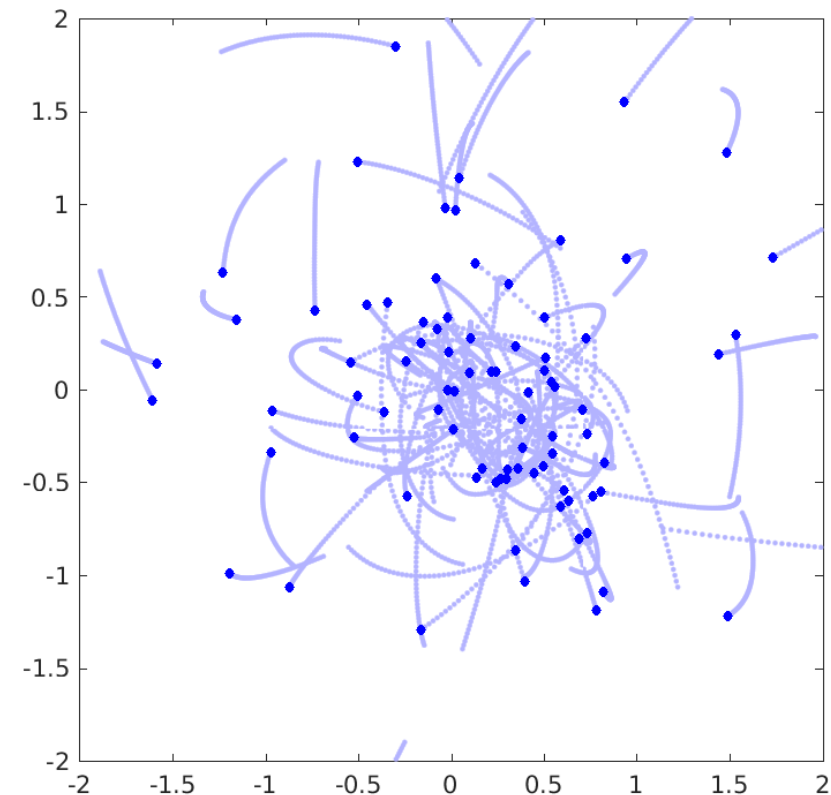
<https://scipython.com/blog/wa-tor-world/>

# nBODY - ПАРАМЕТРИ

- “видимо” двумерно пространство 1K:1K (координатите по X и по Y са от 0 до 1023) – дискретно, но представящо координати с по-голяма точност
- тялото се визуализира в най-близката дискретна позиция
- началните вектор-скорости се генерират предварително рандомизирано в масив с X- и Y-компонента, със средни стойности 1/100 от размера 1024 – значи  $10 \text{ Сек}^{-1}$
- $n = 128, 256, \dots$  – варираме C (изчислителната сложност)
- масите???? (проучете от източници, както и аз)
- брой стъпки на моделното време – 600 за анимация от 30 сек.
- тялото и всяка позиция се представят квадрат 4x4 px
- прилага се директния метод тяло-към-тяло с пълно изреждане (други методи оценяват първо влиянието и интегрират (сумират) само за телата с по-малка дистанция и/ли по-голяма маса
- засичаме полученото ускорение, но не можем да искаме линейност

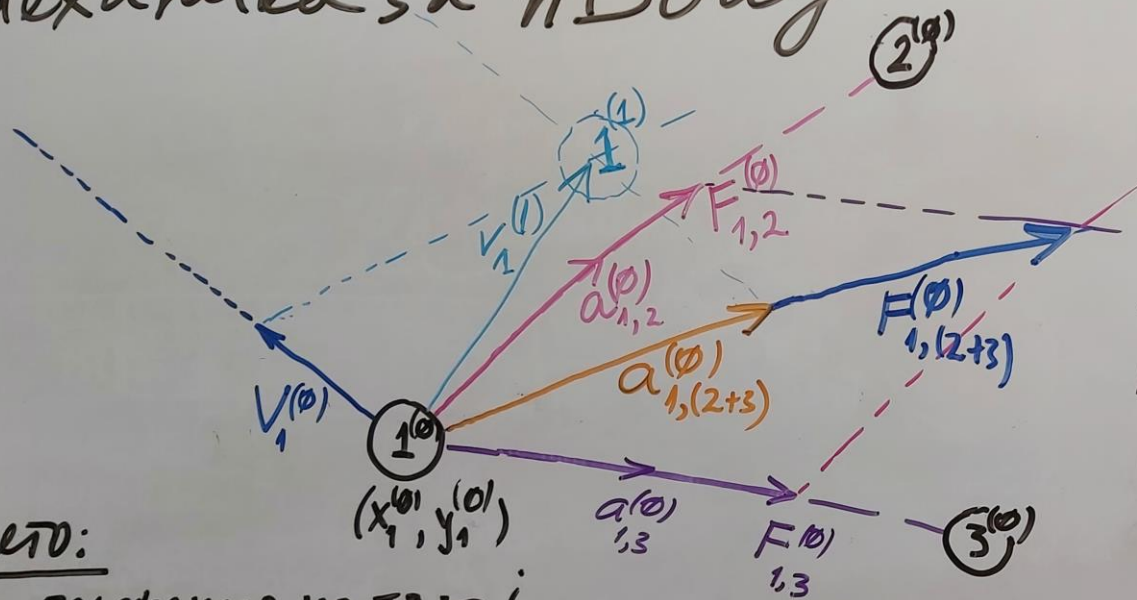
<https://medium.com/swlh/create-your-own-n-body-simulation-with-matlab-22344954228e>

РИТАрх/СПО/РСА – 3. л-я



# ПОЗИЦИЯ НА ТЯЛО В nBody СЛЕД 1ЦА ВРЕМЕ

механика за nBody



$$\vec{a} = \frac{\Delta \vec{v}}{\Delta t};$$

$$\vec{F} = m \cdot \vec{a}$$

$$\vec{F} = m \frac{\Delta \vec{v}}{\Delta t} = m \frac{\vec{v}^{(1)} - \vec{v}^{(0)}}{\Delta t}$$

Така

$$\vec{v}^{(1)} = \vec{v}^{(0)} + \frac{\vec{F}^{(0)}}{m} \Delta t$$

$\uparrow$   
 $a^{(0)}$

$$\Rightarrow \vec{v}_1^{(1)} = \vec{v}_1^{(0)} + a_{1,(2+3)}^{(0)} \Delta t$$

Т.е. за 1 време:

$$\vec{v}_i^{(k+1)} = \vec{v}_i^{(k)} + a_{i(\Sigma n-1)}^{(k)}$$

за тяло i

където:

$(i^{(k)})$  положение на тяло i в момент k

$\vec{v}_i^{(k)}$  вектор-скорост на тяло i в момент k

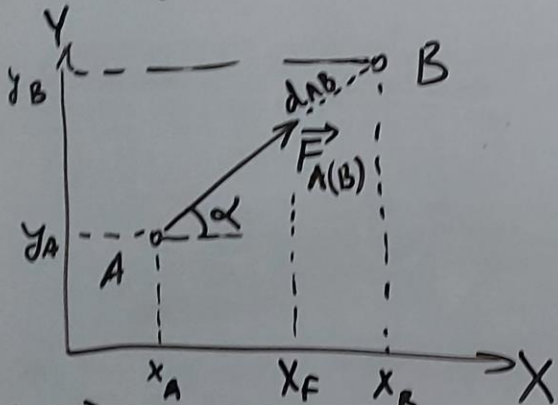
$\vec{F}_{i,j}^{(k)}$  вектор-сила на гравитацията в тяло i от тяло j

$a_{i,j}^{(k)}$  вектор-ускорение в тяло i от гравитацията на тяло j в момент k

# ВЕКТОРНА СУМА В КООРДИНАТИ ЗА nBody

Аналитична геометрия за nBody

вектор-сила от гравитацията на тяло B в/у тяло A:



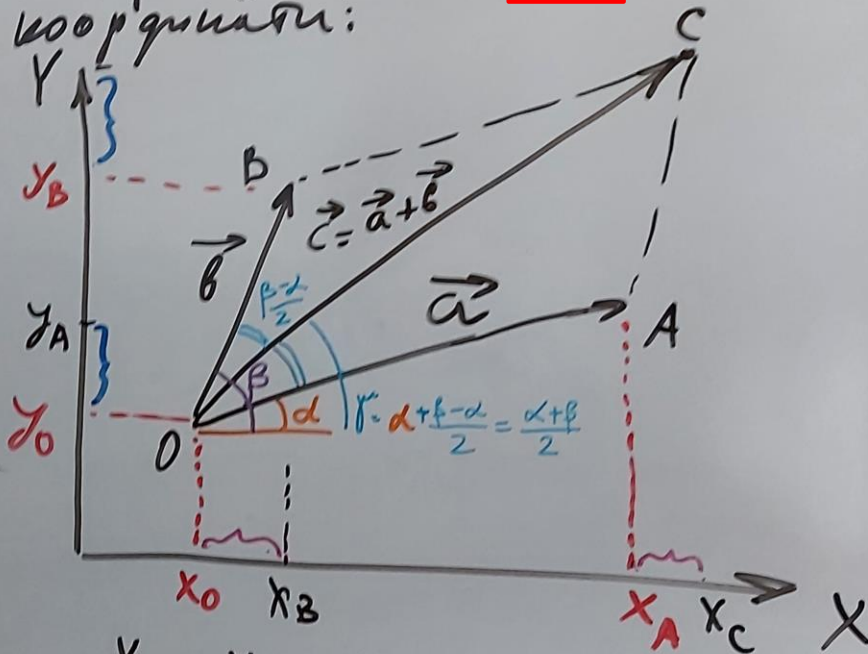
$$|\vec{F}_{A(B)}| = \frac{G M_A M_B}{d_{AB}^2}, \text{ където}$$

разг.:  $d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$

$$\cos \alpha = \frac{x_B - x_A}{d_{AB}}; \sin \alpha = \frac{y_B - y_A}{d_{AB}}$$

$$x_F = x_A + |\vec{F}_{A(B)}| \cdot \cos \alpha, y_F = y_A + |\vec{F}_{A(B)}| \cdot \sin \alpha$$

векторно сумиране чрез декартови координати:



$$x_C = x_A + x_B - x_0; y_C = y_B + y_A - y_0$$

$$\gamma = \frac{\alpha + \beta}{2}$$